CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

FACULTY OF ECONOMICS AND MANAGEMENT

DEPARTMENT OF INFORMATION TECHNOLOGIES

DIPLOMA THESIS

PROJECTING OF INFORMATION SYSTEM FOR PORTFOLIO MANAGEMENT: A CASE STUDY OF A PHARMACEUTICAL COMPANY

Bc. Aleš Erban

Supervisor: Ing. Miloš Ulman, Ph.D.

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Department of Information Technologies

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

Aleš Erban

Informatics

Thesis title

Projecting of information system for portfolio management: a case study of a pharmaceutical company

---

Objectives of thesis

This thesis focuses on the projecting of Informations Systems. The main task will be to analyze requirements for a Portfolio Management System of a pharmaceutical company, propose solution and create comprehensive project plan for the solution implementation, including costs and time schedule. Partial goals of the thesis will be to study and review literature for best practices in capturing and analyzing software requirements and Business analysis, analyze most used project management methodologies and decide which is best to use for accomplishment of the main task of this thesis.

Methodology

Methodology of this thesis will be based on specialized information resources. Process analysis will be conducted using Business Process Management Notation (BPMN). For the purpose of data analysis Entity Relationship (ER) diagram will be created.The Portfolio Management System will be designed using Unified Modeling Language (UML). Project management methodology and Information system development methodology will be used according to conclusions of the theoretical part of this thesis. Based on the literature review and results of analysis and implementation, recommendations and conclusions will be formulated.

The proposed extent of the thesis

60 – 80 pages

Keywords

Project management, Best practices, Capturing and analyzing software requirements, Business analysis, Information system, Analysis, System design, UML, BPMN, Portfolio management

---

Recommended information sources

CHEN Anthony, Beatty JOY. Visual Models for Software Requirements, Microsoft Press, July 2012, 480 p. ISBN: 9780735667723

LANO, Kevin. UML 2 semantics and applications. Hoboken, N.J.: John Wiley, c2009, xv, 400 p. ISBN 04-704-0908-8

MULVEY, Paul, Kate McGOEY, and Kupe KUPERSMITH. Business Analysis for Dummies. Hoboken, NJ: John Wiley & Sons, 2013. ISBN:1118510585

PODESWA, Howard. The business analyst's handbook. Boston, Mass.: A part of Cengage Learning, c2009, xix, 411 p. ISBN 15-986-3565-4

PORTNY, Stanley E. Project Management for Dummies. Hoboken, NJ: Wiley Pub., 2010. ISBN: 1118497236

Expected date of thesis defence

2015/02 (February)

The Diploma Thesis Supervisor

Ing. Miloš Ulman, Ph.D.

Electronic approval: 10. 3. 2015

Ing. Jiří Vaněk, Ph.D.

Head of department

Electronic approval: 11. 3. 2015

Ing. Martin Pelikán, Ph.D.

Dean

Prague on 24. 03. 2015

Declaration

I declare that I have worked on my diploma thesis titled " PROJECTING OF INFORMATION SYSTEM FOR PORTFOLIO MANAGEMENT: A CASE STUDY OF A PHARMACEUTICAL COMPANY" by myself and I have used only the sources mentioned at the end of the thesis. As the author of the diploma thesis, I declare that the thesis does not break copyrights of any third person.

In Prague on 27.03.2015 _____

Acknowledgement

I would like to thank Ing. Miloš Ulman, Phd. and the consultants from the company with which cooperation this thesis has been developer for their advice and support during my work on this thesis.

PROJECTING OF INFORMATION SYSTEM FOR PORTFOLIO MANAGEMENT: A CASE STUDY OF A PHARMACEUTICAL COMPANY

PROJEKTOVÁNÍ INFORMAČNÍHO SYSTÉMU PRO ŘÍZENÍ PORTFOLIA: PŘÍPADOVÁ STUDIE FARMACEUTICKÉ SPOLEČNOSTI

SUMMARY

This thesis deals with collection, analysis and documentation of requirements on a software and projecting the plan for the system creation.

In the theoretical part review of best practices and tools for capturing and analyzing software requirements and Business analysis will be conducted. Afterwards the main project management methodologies will be analyzed, following which, with regard to the practical task, will be decided which one is most suitable to use for that type of projects.

In the practical part of this thesis chosen techniques for reoquirements gathering and analysis will be used and with regards to the results of the comparison of the project management methodologies a project plan will be developed.

The result of this thesis will be a completion of a preliminary study of a product portfolio management system for pharmaceutical company, which will serve for the management of the company as a basis for their devision whether to invest in the development of the system or not.

KEYWORDS

Project management, Best practices, Capturing and analyzing software requirements, Business analysis, Information system, Analysis, System design, UML, BPMN, Portfolio management

SOUHRN

Tato diplomová práce se zabývá sběrem, analýzou a dokumentací požadavků na software a projektováním plánu vývoje software.

V teoretické části bude přehled nejlepších praktik a nástrojů pro sběr a analýzu požadavků a Business analýzu. Po-té budou analyzovány nejpoužívanější metodiky pro projektové řízení. S pohledem na praktickou část bude následovat porovnání těchto metodologií a rozhodnutí, která se nejlépe hodí pro případ projektu v praktické části.

V praktické části budou techniky pro sběr požadavků popsané v teoretiké části aplikovány a s ohledem na výsledek porovnání metodik bude vytvořen projektový plan.

Výsledkem práce bude vytvoření prvotní studie systému pro řízení portfolio produktů pro farmaceutickou firmu. Tato studie bude sloužit management firmy jako základ pro rozhodnutí, jestli do projektu vývoje navrženého řešení investovat prostředky a čas nebo ne.

KLÍČOVÁ SLOVA

Projektový management, Nejlepší praktiky, Zachycování a analýza požadavků na software, Business analýza, Informační systém, Návrh systému, UML, BPMN, Řízení portfolio

Table of Contents

1 Introduction

1.1 Scope of the thesis

The central topic of this diploma thesis is the stage, of the software development lifecycle, till the actual development starts. The stage consists of business and process analysis of a client company as well as analysis, capturing and management of requirements for software development. This means some 40 % of the effort in the software according to an article from toolbox.com (1). While this figure highly depends on the scope of the software, the truth remains that many software development projects fail because of having underestimated this part of the lifecycle.

Besides those activities of the software development lifecycle, another crucial activity of the process is targeted in this work – the Project management. Project management is an activity performed by the project manager during the whole software development project. In this thesis the project management will be conducted up to the point when the solution development will be prepared to start. Then the project plan, including the necessary budget and detailed time schedule will be created for the remaining part of the project – development, testing and implementation.

1.2 Motivation for having chosen the topic

This topic have been chosen, principally because the previously mentioned parts are very tempting area for me. In fact I work as an IT Project Manager in a Pears Health Cyber s.r.o., InPharma division – a company that focuses on consulting and SW development specifically for pharmaceutical industry. In my role I am responsible for leading the whole software development, from its initiation to its completion and handing over to our clients. I am also an analyst – responsible for business and process analysis, requirements collection and analysis and thereafter project plan, budget, precise time schedule and detailed assignment for the programmers.

Having been contacted by Business Development and Portfolio Manager of a pharmaceutical company, seeking to analyze the possibility to create a Portfolio management system and prepare calculations for it, I have decided to exploit that opportunity and use it as the main and central topic for my diploma thesis.

Having done a few projects I realized that the quality of requirements analysis is very critical for further stages and project management needs to be as precise and constant as possible to make the whole development process as smooth as possible. By picking a topic aimed at these activities I seek to use the abilities in those areas I have so far acquired and, mainly, to improve them. I expect to expand my professional knowledge, which thereafter I will be able to use in other upcoming projects. I firmly believe that outcomes of my thesis will facilitate improvement of analyzing the client needs as well as project management and execution in our department and will enable us, in case of all future projects, to deliver good quality software with demanded functionality, in allotted time and within given budget. This work should also be a reference and overview of all main techniques software project and requirements management for all those partaking on software application development.

2 Objectives and methodology

The objective of the theoretical part of this thesis will be in fact threefold. First, review of the literature, articles and studies of best practices for capturing and analyzing software requirements and Business analysis will be conducted, as a result of which our know-how in those areas will have expanded. The subject of the research will also be focused on typical mistakes and deficiencies, which are often brought about during these activities. The know-how then will be applied in the practical part - analysis and projecting of a portfolio management system for a pharmaceutical company.

Then main project management methodologies will be analyzed, after which, with regard to the practical task, will be decided which one is most suitable to use for that type of projects.

The objective of the practical part of this thesis will be to analyze portfolio management processes and needs of a pharmaceutical company and, on top of that, to find a solution that will support these processes. That will include gathering requirements for such a system. In order to receive the most relevant information possible and the most important user's point of view, the processes, the system and other important information will be consulted with a person on a position called Portfolio and Business Development Manager of the client company. Since some sensitive information might be talked about, the name of the person as well as the name of the company will be not written anywhere in this work. As the position name suggests, the person with whom the consultation sessions will be conducted, is a person responsible for portfolio management and business development processes for both Czech and Slovak markets. Sessions with that person will be held presumably once a week, in order to ensure maximally detailed description and analysis of the company's situation, needs and processes.

Once the requirements, process and business analysis will have been done, a solution will be designed which will include complete specification of the system using UML and BPMN diagrams and picking technology base for its creation. Afterwards a complete project plan will be elaborated, including budget necessary for the task accomplishment, team members, time

schedule, WBS and other project management tools. The project picked in the theoretical part will be used and followed throughout the practical part.

The conclusion of the thesis will sum up the gained know-how and knowledge and will evaluate the level of success and eventual problems and other experience to which the work on the thesis will give rise. In the conclusion the accomplishment of the objectives specified in the introduction of the thesis will be commented and evaluated.

2.1 Used methods

For accomplishment of this task standard methodologies for project management, process analysis, data analysis, system design and SW development will be used. Process analysis will be carried out using Business Process Management Notation (BPMN). For the purpose of Data Analysis Entity Relationship (ER) diagram will be created. The design of the Portfolio Management System will be proposed using Unified Modeling Language (UML). Project management methodology and Information system development methodology will be used according to conclusions of the theoretical part of this thesis. Based on the literature review and results of analysis and implementation, recommendations and conclusions will be formulated.

2.2 Assumptions and restrictions

Among main assumptions of the successful accomplishment of the objectives of this work belong active participation of the Portfolio and Business Development Manager as well as possibility to consult with other members of the project team and full access to all the tools that will be needed.

The biggest restriction of the work regards the character of the information gathered from the consultant in the practical part. As the information can be of potentially a sensitive and confidential character, it must be treated as such.

Names of employees or systems of the client company as well as the company's name itself will have to be concealed.

3 Theoretical part

3.1 Software requirements and business analysis

Requirement definition

First, we need to understand what actually a requirement, if we are supposed to talk about a science based around it. In the literature we can find quite a lot of different definitions. One that is most accurate and specifically targets software requirements is a definition from Karl Wiegers (2), who defined it in his popular book Software Requirements as follows:

"A requirement is a statement of a customer need or objective, or of a condition or capability that a product must possess to satisfy such a need or objective."

Another good and rather simple definition presented Suzanne and James Robertson, in their book "Mastering the Requirements Process" as follows (3):

"A requirement is something the product must do or a quality it must have."

The International Institute of Business Analysis in its guide book called 'A Guide to the Business Analysis Body of Knowledge® (BABOK® Guide)' defines a requirements in two manners as follows (4):

"A requirements is a condition or capability needed by a stakeholder to solve a problem or achieve an objective."

The standard ISO/IEC 24765:2010 defines requirement as follows (5):

"The requirement is specified as a condition or capability that must be met or possessed by a system, system component, product, or service to satisfy an agreement, standard, specification, or other formally imposed documents."

While each of those definitions is written in different words, when reading them carefully we realize they all are very close in meaning to one another. In further context then by

a requirement will be meant anything that in some way specifies qualities or features of future software product.

The primary objective and key success factor of any piece of software is in principle the level to which it fulfills the purpose for which it was developed. No matter which particular software development methodology (will be described and explained later on) is followed, it always has to be clear, what particular purpose is the developed application supposed to serve and what set of processes it has to support. The demanded functionality is expressed by requirements.

Representation, structure and the level of particularity of those requirements differ depending on the character of the project and on the software development methodology used. In so called rigorous methodologies (for example RUP) the requirements are written in detail and modeled using various tools, on contrary in so called agile methods (for example SCRUM) the requirements are expressed only as a few descriptive words, perhaps on a blackboard or are even only in minds of the developers, who are also in those methodologies software analysts as well. Regardless of the particular representation of the requirements, it is necessary to realize their importance and how crucial the right attitude to gathering and analyzing them is.

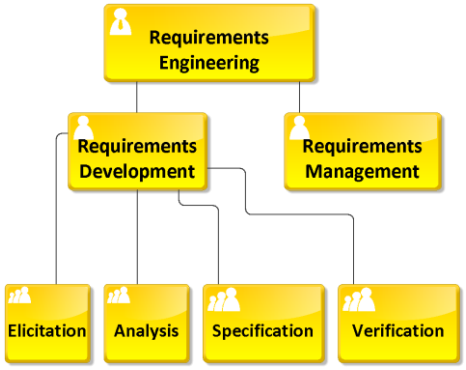
Capers Jones, an American specialist in software engineering methodologies (6)wrote that up to 50 % of errors in the software development are given rise already during the phase of the requirements gathering and designing an application. Another great figure in the software engineering field – Steve McConnel, in his book Code Complete (7), alleged the number goes up to 60 %. Furthermore, an error discovered in the phase of analysis of requirements is roughly one hundred times cheaper to amend than an error occurred in later implementation phase and even more than if the error is discovered during operation of the piece of software. Capers Jones also claimed that correction of errors in software having origin in requirements themselves typically make up to 50% of all costs on software development. Yet, for the phase of capturing analyzing the requirements is on average allotted only 12% of overall timespan of a software development project (8). While these figures are very dependent on the type and size of the project and their accuracy is therefore debatable, the main principle here is apparent – the sooner an error or a flaw in a software is discovered, the less expensive and time consuming it is to fix

it. The phase of capturing and analyzing the requirements and business process analysis is hence exceedingly important and it is therefore crucial to pay enough attention to it at any project.

The scope of the Requirements Engineering

The discipline of capturing, analyzing and managing requirements can be divided into more sub disciplines, as did Karl E.Wiegers in his book Software Requirements (2). What he did was that he divided the discipline into several bounded fields, as is shown on the following diagram.

Figure 1: Software analysis disciplines (2)



Wiegers calls the whole discipline as a "Requirements Engineering" and divides it into two sub disciplines – "Requirements Development" and "Requirements Management". Within the "Requirements Development" discipline he distinguishes four activities – Elicitation, Analysis, Specification and Verification. We are going to look into those categories more and see what's what.

Requirements Development

The term "Requirements Development" might be a little bit misleading. The requirements are, of course, not developed, the requirements for a software product are already existent, respectively appeared as a result of some needs. Usually the requirements are in fact initially in

minds of the future users and sheltered in the business processes of the client company. It is just necessary to discover them, elaborate them and document them in such a way that they can be turn into a supportive and functional software product. And by Requirements Development is meant exactly the process of retrieving the requirements from users' minds and business needs. In particular it means such activities as identification of the future user groups of the software and assuring representatives from each of them, getting to understand individual tasks the software is supposed to do and business processes it has to support, then gathering and analyzing all the information from potential users, distinguishing the functional and nonfunctional requirements, separating relevant information such as business rules from unnecessary redundant information and prioritization and validation of the requirements, including the control of their comprehensiveness and relevance regarding defined objectives.

Requirements Management

Requirements Management is a discipline which aim is to ensure compliance between the client and the developers upon the requirements. It basically means transforming the requirements into clear assignment for developers while assuring the developers correctly properly understand what they need to do so that the software meet the requirements. For that corresponding documents and models need to be created so the communication between users and developers is unambiguous. Among particular activities of this discipline belong transformation of the information gathered from the client consultants into a formal definitions and assignment for programmers, in case of requirements for change assessment of the impact and endorsement of the suggested changes, managing the endorsed changes when implementing them into the project, but also adjusting the project plan according to actual requirements and circumstances, negotiating about the conditions and obligations of the implementation of the changes and their expected impacts, observing the effects of individual requirements on the software solution design, its source code and testing scenarios. It also means controlling the state of the requirements during their life cycle.

In summery then, the Requirements Development is a process whose central interest is capturing and listing each individual requirement. The Requirements Management on the other

hand concerns itself with the managing the transition of those requirements into a formal document, models and assignment for developers as well as managing already defined requirements and their changes during the software project lifecycle.

3.2 Types of requirements

The basic division, that was already mentioned, is on functional and non-functional. The difference is that a functional requirement represents a direct functionality of a software system. Basically it is a statement expressing what the system should do, what should be its behavior, how should it react to certain inputs and what should be the corresponding outputs. It can, however, be also a restrictive statement – saying what the system shouldn't do.

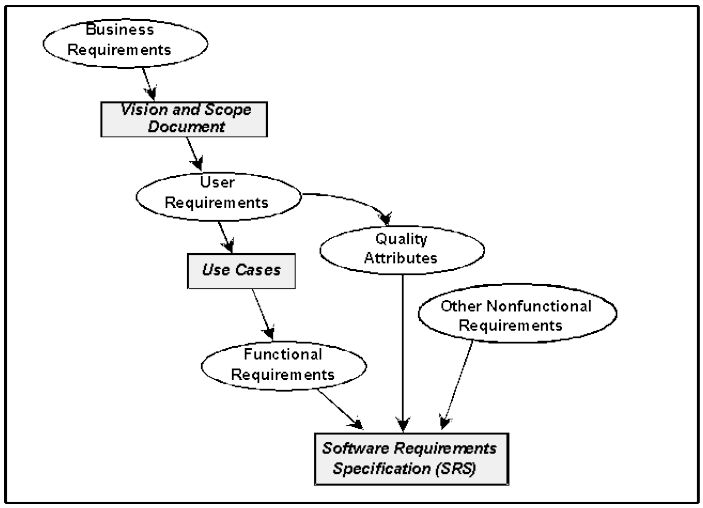
The non-functional requirements are simply any other requirements. It means such statements defining some sort of quality or attribute the system needs to have once developed. Achieving or not may make difference between brilliant everyday used and forgotten unused app, only representing some (and probably not small) expenses. The requirements in this category concerns the system's security, graphical design, legal restrictions, but as well requirements on the technology used and architectural limitations and demands. Later mentioned requirements would be those expressed by technical stuff of the client company.

There is also another type of requirements - requirements regarding the project itself. Suzanne and James Robertson call those 'restrictions'. Those are 'restrictions' concerning the deadline of the project or the budget, but also demands for doing a legal research for example before implementing a certain functionality (from my own experience).

Karl Wiegers (9) divides the requirements into categories also depending on which phase they are in the requirements capturing process. First come "Business requirements", it means those that arose from a business need. From those an initial document, which he calls Vision and Scope is created and which content is clear from its name. After that comes the "User requirements", those are requirements collected from potential user gathering information about what might individual users need. According to this part of the analysis, Use case models (will be described later on) are created. Those Use case models then represent the functional

requirements. Besides them there are also some “Quality attributes” of the system to be developed and the non-functional requirements from previous paragraph. Put all together, the documents of these categories of requirements constitute a document by Wiegers called Software Requirements Specification.

Figure 2: Requirements according to phases (9)



In Pears Health Cyber we divide the requirements in very similar manner. However, both functional and non-functional requirement are usually divided into more categories, to increase the transparency of the whole requirements analysis.

3.3 Software analyst

The job

The person in charge of the activities described above – i.e. requirements capturing, analysis, modeling and documentation, business processes mapping, analyzing, modeling and documentation as well as ensuring the proper understanding of those by developers, is usually called a software analyst, sometimes called business analyst. Of course the role can be connected

to other roles to create one vocation. In the Pears Health Cyber firm, for example, the Software/Process analyst is together with a project manager role, creating together an IT Project Manager position. On the other hand, in some companies especially those that work very often on bigger projects, the role is actually diversified into more roles such as IT analyst – the one discussing more technical issues, business analyst – the one who needs to understand the business the client company is doing, or process analyst – the one mapping the processes of the client company. Together then they put together the specification of the system needed, in case of the bigger projects means more than just creation of one piece of software, but also optimization of the business processes, integration of more systems together and so on.

It need to be said, that the task of the Software analyst is not only to collect requirements. His task is basically to put himself into the situation where the client. He can't only try to support and make more efficient the workflow of the client's current system, he needs to thing about new workflows and processes that would make it even more efficient than just a new software. The software is then created to support the new workflow and processes. There is an interesting saying explaining this nicely – "from what the client says he needs, the software analyst has to make up what the client really wants.

Qualities of the software analyst

A software analyst is a person with an extensive set of knowledge and highly developed abilities, who represents a sort of 'bridge' between two utterly different worlds – the world of business and the world of Information technology. Two worlds with completely different nature, culture and language. It is necessary that the analyst is capable of translating between those two worlds, for which he needs knowledge and abilities from both these areas. In some areas the analyst also needs the knowledge of the particular domain in which client conducts his business, such as banking or pharma and biotech, in order to be able to correctly understand the requirements and talk to the professionals from the domain about what they do and what they might possibly need for supporting or enabling it. Being able to 'translate' between the business world and the IT world is then crucial to correctly set the goals of the software and avoid later having to change

something as a result of poor transformation of business need into software functionality, which, as explained earlier on, is later becoming more and more expensive.

We can find a list of particular knowledge of a software analyst in books such as Software Requirements from Wiegers, or the A Guide to the Business Analysis Body of Knowledge (BABOK Guide) from the International Institute of Business Analysis. Here only a few most important and most striking abilities and knowledge domains of a software analyst were highlighted.

- Knowledge of the application domain and technology overview
- Empathy, ability to communicate and give appropriate questions
- Ability to listen and understand what is "between lines", what the user really needs though he is not explicitly expressing it
- Good written as well as spoken word
- Abstract and complex thinking, ability so observe reality as a whole and abstract from details
- Managerial and organization abilities, ability to make decisions and prioritize, punctuality, assertiveness, ability to negotiate, proactive approach, creativity
- Knowledge of tools and techniques for business and requirements analysis
- Possession of systematic and critical thinking

3.4 Tools and techniques for requirements analysis

Software projects are quite often plagued by incomplete, unclear, or even omitted requirements. As a result of bad requirements practices, many projects are doomed to fail. There is also usually quite a long distance between the initial requirement, which tends to be very general, indeterminate and vague and the final, well documented, specified, clear and unambiguous requirement. The way nonetheless has to be undertaken in case of every single requirement in order for the requirement to be correctly understood by all stakeholders. Software analysts have been realizing that reality and have hence throughout the years developed many tools, techniques, guidelines to tackle this issue and enable future analysts to achieve well

documented, clear and unambiguous requirements more easily and quickly. Of course, each requirement, situation and case need a different tools and moreover each software analyst may prefer a different set of tools. The software analyst nonetheless must be able to pick the right tools and techniques and use them appropriately so as to maximize understanding the requirement.

The most known and used tools and models will be now reviewed and described in order to acquire an overview of them. Note that comprehensive description of the models is beyond the scope of this thesis and interested reader should navigate to the references used for more ample information.

Visual models for requirements documentation

A visualization of requirements is very important. A common practice when requirements are simple written in a document one after another as a list with a very little structure is very difficult to grasp and analyze. The following table illustrates such a list. Now if we imagine that there would be hundreds in bigger systems thousands of lines, one can see the point.

Table 1: Example of requirements list

Requirements list	
REQ001	System shall have fields for title, first name and last name.
REQ002	System shall display a name if there is one in the stored profile.
REQ003	System shall require name is completed.
REQ004	System shall have a field for position or title.
REQ005	System shall require that position field is not empty.

Source: Author, 2015

Agile development methods usually use product backlogs, user stories, and acceptance criteria, which is a bit more structured, but it is still in written form and the expressive power is still insufficient. There is simply no way the human brain can make sense of the complexity unless it is broken into smaller organizational groups. Human brain also remembers and can work much

better with a visual information then with a text. There is a saying “a picture is worth a thousand words” and it seems appropriate when it comes to software modeling (not only requirements). Models are basically a visual representations (images and pictures) of an information related to the processes, data, and interactions which are to be performed and processed in the system being developed. People use such models every day without even realizing it.

3.4.1 Introduction to RML and UML

UML (Unified Modeling Language) is a language used to visually specify the design of software systems (Object Management Group 2007). UML is therefore not specifically aimed at requirements modeling, but rather at modeling the software’s architecture and technical design. There are nonetheless some models defined in UML specification that are concerned with requirements modeling as the requirements are at the end of the day an essential part of the software architecture as well, since the software’s architecture and design are modeled as a response to the requirements. The UML includes following models:

- Class diagram
- Object diagram
- Use Case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

From those only the Use Case diagram is concerned directly with requirements and activity diagram and State chart diagram can to some extent be used in the requirements analysis. The class diagram is then used by analysts to figure out the data structure of a system. The rest of the diagrams, however, is mainly for software architecture and technical design. UML then is a

reasonable foundation for requirements modeling, nonetheless it is incomplete for modeling requirements because it lacks models that connect requirements to business value and models that present the system from an end user's point of view (10). UML is meant to be used to describe the technical design and architecture of a system, which enables further communication between the software analyst and developers. For requirements modeling it is, however, insufficient and its models need to be extended. As a response for that need a RML language came along described in the book Visual Models for Software Requirements (10). The book defines the RML as follows:

"RML (Requirements Modeling Language) is a language designed specifically to visually model requirements for easy consumption by executive, business, and technical stakeholders. RML is not a theoretical modeling language. In developing RML, we modified existing models for ease of use and created new models to address gaps. The result is a full set of models that, at its core, is specifically designed to model software requirements and is easily adopted by business stakeholders who are often challenged by complex models."

Figure 3: Comparison of RML and UML (11)

Modeling Characteristics	RML	UML
Understood by Business	✓	✗
Understood by Developers	✓	✓
Understood by Testers	✓	✓
Maps to Business Objectives	✓	✗
Includes KPI's	✓	✗
Includes End User Viewpoint	✓	✗
Easily Adaptable	✓	✗
Simple to Teach	✓	✗
Software Concentration		✓
Requirements-focused	✓	✗
Standards for Syntax and Semantics	✓	
Visualizes System Architecture	✗	✓
Visualizes Database Schemas	✗	✓

The table above shows a comparison of the UML and the RML in terms of their attributes and usability in various cases. Apparently an analyst should know models from both in order to be able to communicate with both the business personnel (client) and other, technical participant of the software development project.

Some of the most important and most widely used models will be now reviewed. For better readability, the models have been divided and put in groups as in the Visual Models for Software Requirements book (10) - objectives models, people models, systems models, and data models, known collectively as OPSD.

Figure 4: OPSD classification of RML models (10)

Each of those categories has a bounding model. It is a model which with high probability contains all necessary information for the category. The bounding models for each the categories of RML are the Business Objectives Model (objectives), Org Chart (users), Ecosystem Map (systems), and Business Data Diagram (data). For instance all possible stakeholders of the system can be identified from a corporate organizational chart, which can then be used to create a solution Org Chart. Therefore, a complete list of relevant stakeholders to be interviewed is determined. The following table describes the model categories and provides the list of models within them (complete list of the models is available in (10)). Some of the most important models will be more thoroughly examined afterwards.

Table 2: RML model categorization

	Description	Models	Bounding model
Objectives	Describe the business value of the system and help you prioritize features and requirements based on their value	Business Objectives Model Objective Chain Key Performance Indicator Model Feature Tree Requirements Mapping Matrix	A Business Objectives Model bounds the objectives space
Users	Describe who is using the system, along with their business processes and goals	Org Chart Process Flow Use Case Roles and Permissions Matrix	An Org Chart bounds the people space
Systems	Describe what systems exist, what the user interface looks like, how the systems interact, and how they behave	Ecosystem Map System Flow User Interface Flow UML Class diagram Display-Action-Response Decision Table Decision Tree System Interface Table	An Ecosystem Map bounds the systems space
Data	Describe the relationships between business data objects from an end-user perspective, the life cycle of the data, and how that	Business Data Diagram Data Flow Diagram UML Class diagram Data Dictionary State Table State Diagram Report Table	A Business Data Diagram bounds the data space

	data is used in reports to make decisions		

Source: (10)

Objective models

3.4.1.1 Business Objectives Model

The main purpose of a software project is to bring value to the business, to its users. To keep the requirements for a software system in line with the business objectives it is a good practice to model the business objectives first and upon them to come up with requirements for the system, in order to avoid creating such features that do not contribute to the business and its objectives.

In the business objectives model every item consists of four elements: Business Problems, Business Objectives, Solution Concept and Success Metric. These are defined in the following table:

Table 3: Business object model elements

Element	Definition
Business Problem	Issue preventing the business from achieving its goals.
Business Objective	Measureable target that specifies when the business problem is solved.
Solution Concept	Vision of the actual solution that the business chooses to implement in order to meet the business objective. It is typically described by a list of high-level features.
Success Metric	A business objective that will actually be measured to determine whether the project is successful, or additional measures that are related to the solution.

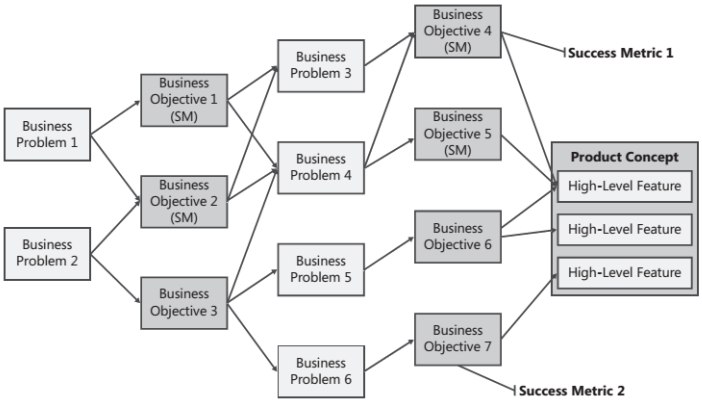
Source: (10)

Business Problem then gives rise to Business Objectives, which means an initiative for a change. The new Business objective then creates the need for a solution – a Solution Concept in the form of high-level features of the future system is proposed. The Success metric is then the ability of the system to support and enable the business objective. This dependency of the four phenomena is illustrated on the following diagram:

Figure 5: Business Objectives Model Elements

Note that newly occurred Business Objective can bring along another Business Problems. That means that Business Problems and Business Objectives can create a chain. Every Business Objective then has to lead to another Business Problem, or to the Solution Concept, respectively to the High-level feature which supports the objective and hence solves the Business Problem. An example of the Business Objectives Model is shown on the following diagram:

Figure 6: An example of a Business Objectives Model template (10)



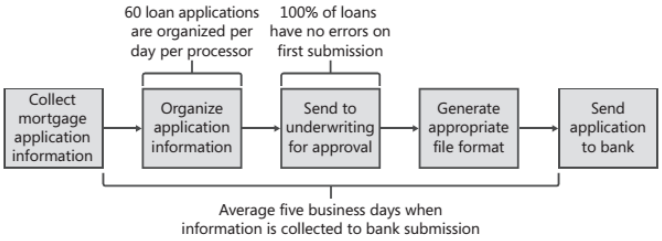
3.4.1.2 Key Performance Indicator Model

Every project should have set of Key Performance Indicators. A Key Performance Indicator (KPI) is a metric, which indicates the level which a project (or generally an investment) or its part needs to achieve to be regarded successful. Those KPIs can be extended Success Metrics, however are more particular. KPIs also map to the business processes, but explicitly say what performance is to be achieved. While Success Metrics can have a form of a simple sentence, saying the system supports certain process, for instance, a KPI will include particular numbers and figures so, once project is finished, it is possible to clearly say whether the goal was achieved or not. The KPIs are usually used in case of projects where the added value must be apparent, measurable and proven to the investors or sponsors (more about roles in projects in chapter Project Management).

Of course, it not always simple or even meaningful to measure success of a software system, when it is replacing an old one. The problem there is that there is usually a mismatch between the contemporary processes and new system. That usually leads to proposal of new processes and hence it is not only the system that should be measured but the solution as a whole (including the new processes). The KPIs need then to be created with that in mind.

The KPIs can be written simple as a list of items (sentences), or can be included in the process diagram (described later on). An example will show and explain this best. Let's imagine we have a financial institution which would like to replace its contemporary solution for loan origination system for a new one. Now the contacted software supplier company has estimated that the change of the systems will bring saving of approximately €10 million a year. In that case the €5 million price tag for the solution seems to be a good value. The management of the financial institution, nonetheless, want to be sure these figures will be actually achieved. Which is why there will have to be many KPIs metrics to show the added value. Now we can have a look at a few of the KPIs in a process of submission of mortgage application.

Figure 7: A loan origination KPIs example (10)



As showed on the picture the KPIs are added directly to the process, where every KPI is assigned to one or more activities in a form of a statement with a certain figures explicitly setting the goal for the solution.

3.4.1.3 Feature Tree

Very useful for overall overview of the features of a software solution is the Feature Tree model. The model shows the structure of the features of a software and puts them in logical groups, which depicting the systems' scope on a single page.

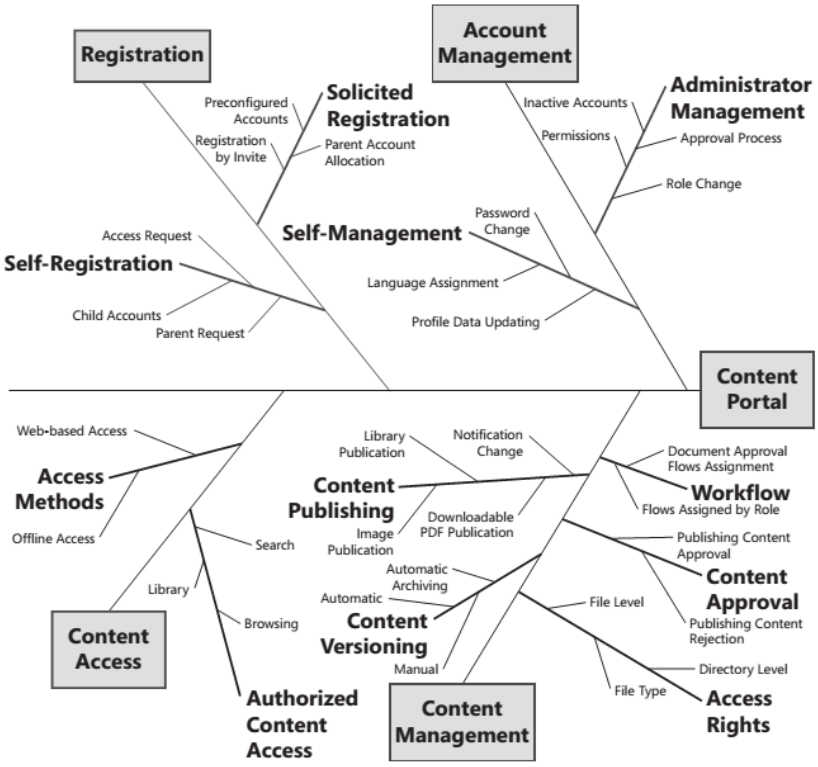
At this point we need to define what exactly is meant under the word "feature", to be clear what will be used in the model:

"A feature is a short-form description of an area of functionality that the solution will ultimately include to meet the business objectives .Features are collections of requirements that are used to articulate and organize the requirements." (10)

The model takes the form of a graph similar to one of a mind map and looks like the fishbone diagrams (sometimes called ishikawa diagrams). At the center of the graph is a start node, from which several branches come, each representing a top level feature and at the end of which is the name of the feature. Any of the branches can then be expanded for other lover level branches of features and so on recursively, altogether showing all the features of the new system.

Once again an example will show the idea best. Let's imagine we have a company creating a web portal. The portal has some restricted pages and hence needs its user to authenticate before accessing them. The following diagram is then an example of the Feature Tree diagram:

Figure 8: Feature Tree model example (10)



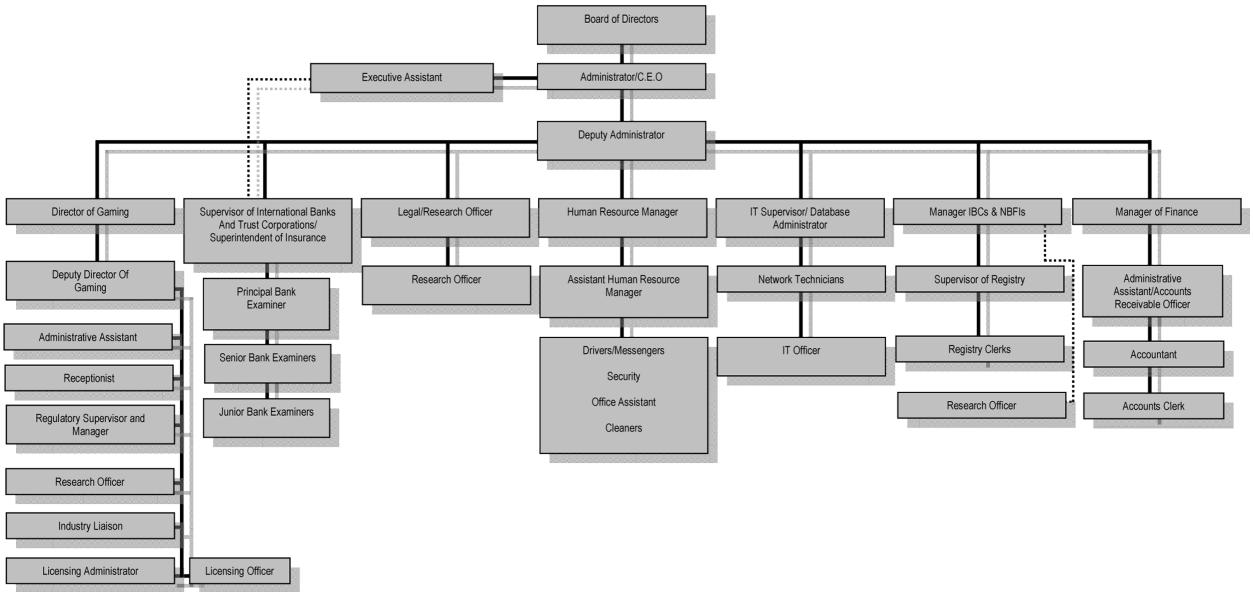
USERS MODELS

3.4.1.4 Org Chart

As a basis for identifying all stakeholders who will be using or will provide input or process output of the developed software serves the Organizational Chart. It is a diagram of all the people in an organization, showing the organization's structure and the people's roles. Many companies already have and maintain such charts and are therefore available for the software

analysts, for which it should be a key document for identifying the key users, which will then determine the requirements for the system. Not all companies, however, maintain comprehensive Org chart with all people in the organization. It is logical as in big companies the chart would be vast. It depends on the project what level of detail is necessary in the Org chart, how "deep" it must be. On the following picture is an example of such a chart:

Figure 9: example of an Org Chart (10)



3.4.1.5 Use Case

Use case diagram is a model describing all the possible interaction of a user with the software system to be developed. Note that the user in this context can be a person, or another software (or even hardware) which uses the output data of the system as its inputs. The model does not say anything about "how" the system will perform given activities, but instead it shows what an actor will be able to do with the system. "Use cases are written to describe how an actor interacts with a solution to accomplish one or more of that actor's goals, or to respond to an event" (4). It means that the Use Cases are independent from any particular implementation. Use cases are

normally more comprehensive than the set of features in the Feature Tree model, which can be taken as a basis for the Use Case diagram creation. The Use Case diagram is very well suited for the definition of the Service Level Agreement (SLA).

There are two representations of the model described in literature. First, the Use cases take on a form of a set of tables. The structure differs from author to author and there don't seem to be one common protocol for the tables' creation. The following fields seems to be most meaningful and common: (instead of actual values an explanation has been provided for each of the fields)

Table 4: Use case table representation

Use Case ID:	This is a unique identifier for the Use case for the purposes of referencing between several Use cases.		
Use Case Name:	The name take the form of a phrase with an active verb. (for example "Create account")		
Created By:		**Last Updated By:**	
Date Created:		**Last Revision Date:**	
Actors:	An actor is a person or other system providing an input or using the outputs of the system, in this particular case one (or more) that communicates with the system via this Use case.		
Description:	There can be a short description of the Use case (its purpose).		
Trigger:	The trigger is an event that initiates the Use case. The event can be initiated by an actor (outside the system), or by the system itself (the system changes its state).		
Pre-conditions:	This is a list of conditions (statements) that must be fulfilled or activities that must have taken place in order for the Use case to be initiated.		
Post-conditions:	This is a list of conditions – a state in which the system has to be once the Use case is finished. The Post-conditions must be fulfilled for all possible executions of the Use case. It is possible to describe here separate post-conditions that are true for successful and unsuccessful flows through of the Use case.		
Normal Flow:	This is the normal and expected "everything successful" flow through the Use case. It is detailed description of the user's actions and the systems responses in a form of a scenario. The individual steps are designated by numbers – it is a numbered list.		

Alternative Flows:	These are the flows of the Use case where a special conditions come true, where an untypical situation occurs. These are designated by numbers as the normal flow and above those scenarios has to be the condition or event that initiated the alternative flow.
Exceptions:	In this field any errors that can be anticipated can be written here along with a reaction of the system to them.
Includes:	This is a list of other Use cases which are called ("executed") by this Use case.
Notes:	Other additional information for the Use case.

Source: author, 2015

The second representation is a Use Case Diagram, described in the UML specification (12). It consists of the following elements (4):

- Name
- Actor(s)
- Preconditions
- Flow of Events
- Post-Conditions
- Relationships

The relationships between actors and Use cases are called associations. An association line indicates that an actor has "access" to the functionality depicted by the Use case. Relationships between Use cases have so called "stereotypes". There are two stereotypes two Use cases can have between each other:

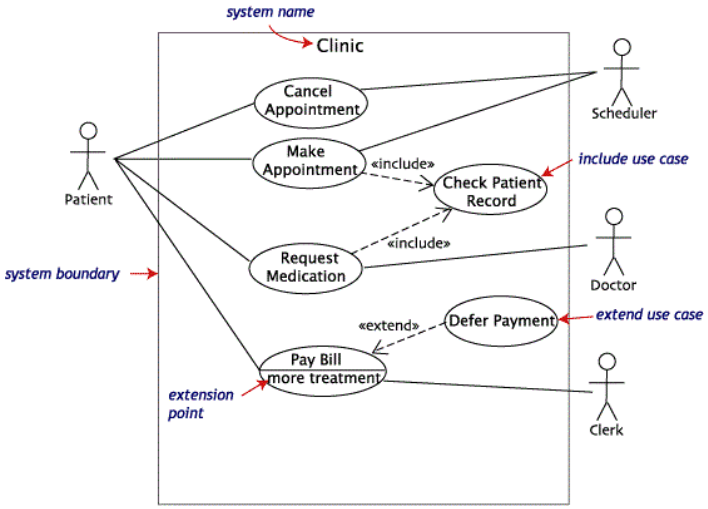
Extend: this type of association is an "extension" of the functionality represented by a Use case being extended. The extended use case, however, is completely functional on his own, so the extending Use cases are not necessary in the flow of the extended Use case. An extension is

effectively what was described in the table version of Use cases as the “alternate flow”, but is captured in a separate use case for convenience and better transparency of the model.

Include: this type of association is an “inclusion” of the functionality represented by another Use case. The included Use case is not necessarily functional Use case itself, if it is not possible to directly execute the Use case by an actor. This relationship is most often used when some shared functionality is required by several use cases in order not to replicate the information in the model.

An example of the UML Use case diagram is shown on the following picture:

Figure 10: Example of the Use case diagram (13)



The table representation is much more thorough and expressive, however it is not convenient for communication of functionalities with the users and even programmers might have difficulties grasping the information from them. While the Use Case diagram might not have sufficient expressiveness, it is visually very clear and the missing expressiveness can be easily amended by adding activity or process diagram to it, together creating basically the same explanation as the scenarios written in the table representation. As everywhere every analyst and his team might prefer different tools and it is up to the project manager of the team to set a common protocol so as the documentation is coherent. In case of bigger systems both these representations are combined together to ensure clear understanding of the functionality of the system by both the users and the developers.

3.4.1.6 Prototyping

Developing a software system is obviously a very complex process. Using the previously mentioned models the software analyst has an idea about what the client organization needs. And now it is time to make suggestions and proposals for the particular functionality of the system and its design. One of the most useful, clear and precise methods for that is called Prototyping. This method is suitable for all systems (or its sub components) where all activities of the system are behind some user input – a screen. It is possible to draw the screens via a mockup tool, in a form of wireframes. These screens then are a basis for the actual design of the system. Or it is possible to directly have the design proposed by a designer and then make the screens almost exactly accurate. Such a model then is very good for communication and amendments of the functionality of the system with the client, because he can see the actual functionality "life" although in fact it is just a nonfunctional mockup – a prototype.

This technique has been used extensively by producers of cars, cell phones and other products, to see the reactions of potential customers, to give them the idea what it will look like and then communicate what could be amended before actual creation of the product. The same principle is valid in case of the software prototypes. As in the case of the consumer products, the actual software can differ from the prototype. The client then needs to be acquainted with this fact to avoid later misunderstanding. Among big advantages of this technique belongs the fact that a user can see first "touchable" result, without actually having invested in the development and so in case a need for change is discovered here, it is not too expensive.

On the following two pictures are examples of the wireframe mockup prototype and a rigorous prototype of actual design of the application.

Figure 11: Wireframe prototype example

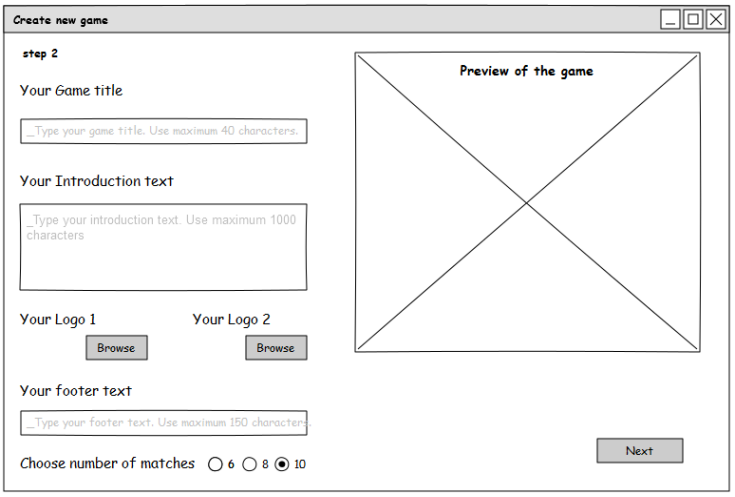
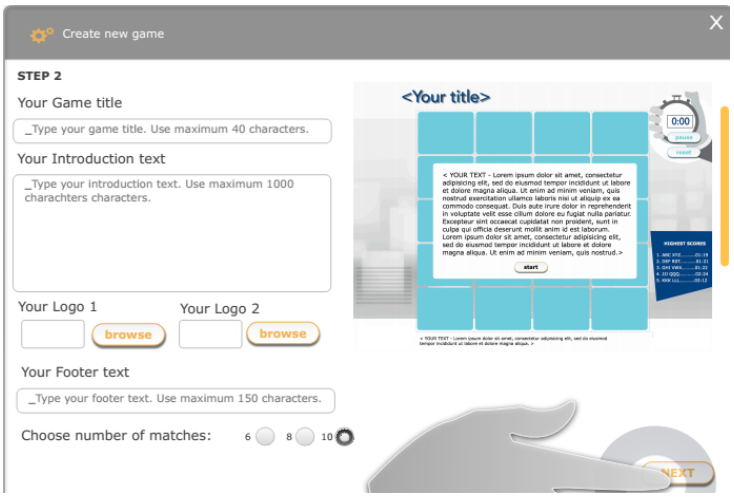


Figure 12: Accurate prototype example



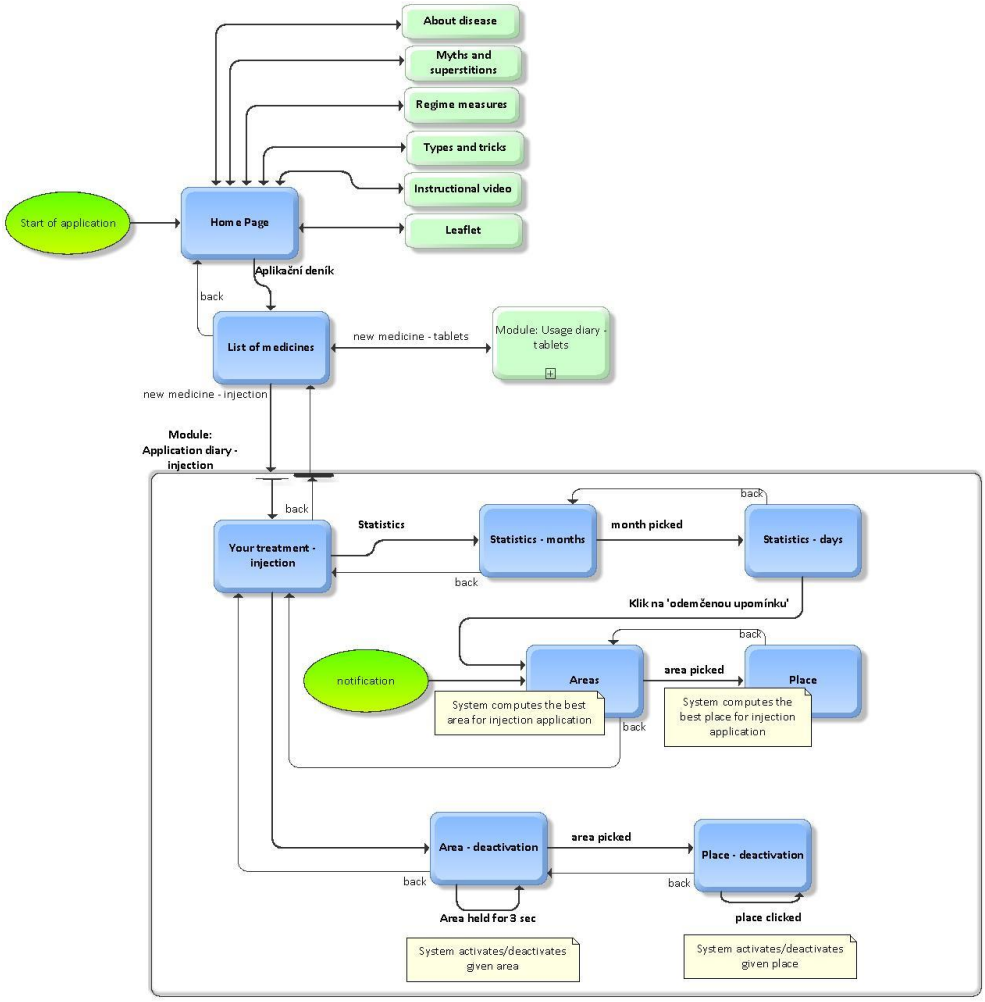
3.4.1.7 User Interface Flow

The User Interface Flow model is a visual representation of the transitions between the screens of the system to be developed. It resembles the Business Process model, except the nodes are

not activities but the individual screens and the transitions are launched by an event, for example clicking on a button. Now it has to be possible to identify each of the screens in the model, for which it is advisable to name the screens the same way as in the previous Prototype model. Together the two models give quite a good overview of the system. In case of smaller projects these two models are often completely sufficient for specification of the software and for communication of features and design with both the client and the developers.

The following example of the User Interface Flow model shows the process of an injection application diary module of an application used for patients taking drugs in a form of tablets or injections. Each of the nodes is a name of a screen, which has been before created as a prototype.

Figure 13: User Interface Flow model example

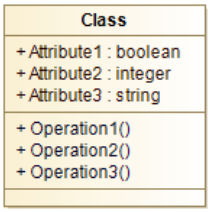


For modeling of the User Interface Flow it is possible to use any process modeling tool, as even basic set of elements for process modeling is enough to create the User Interface Flow model.

3.4.1.8 Class diagram

The Class diagram depicts the classes within a model. Classes have attributes (the object variables), operations (the object methods) and relationship with other classes. The fundamental cornerstone of the diagram is the icon representing a class:

Figure 14: Class by UML definition



As we can see there are three parts in the icon. In the first one there is the name of the class. In the second one there is a set of attributes the class has and in the third part there is a set of operations the class can perform.

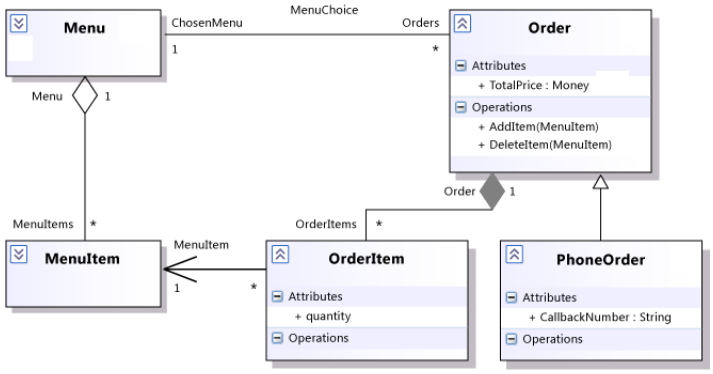
Software analysts can use the Class diagram, specified by the Object Management Group (12) for two purposes. The first one is to use it as a Domain model, where each of the objects modeled represents an abstract view of the relationships between the classes. Such a model is then used for communication with clients and key users who can thanks to the diagram envisage the structure of data which enables them to figure out more easily which data they might want to save and what are the relationship between different objects in the system. The diagram then can be used as a basis for the ER diagram (described later on) and ultimately for the actual database structure of the system. It is possible to put attributes (at least the most important ones) already in the Domain model, nonetheless, for better overview and understanding it is advisable to only model the relationships between the classes in this model and then leave the data structure on the ER model.

The second one is more technical – using the Class diagram as an Objects specification model in an object oriented system. Those classes then can be used as bases for real classes in a programming language. Most commercial tools even support generation of the code from the Class model. The Domain model is then extended for the attributes and operations.

The difference in the two usages of the Class diagram is that while the Domain model is fully independent from actual implementation, chosen language, frameworks and so on, the Object specification model can be platform dependent and is concerning the actual implementation. Besides attributes of the objects necessary for the implementation (which are

not necessary for the Domain model) the classes also includes the methods, defining behavior of the class in the system. The following picture shows an example of a class diagram.

Figure 15: Class diagram example (12)



The full description of the Class diagram, its elements and possibilities is not in the scope of this thesis. The whole specification, along with several tutorials into the Class diagram is available on the web sites of the (12).

DATA MODELING

3.4.1.9 Entity-Relationship diagrams

The ER model is basically a representation of entities and their relationships to one another. An entity is a piece of data – it is an object which data is stored in a database. A relationship is a "status" of one object towards another and represents how the data is shared between those entities

The entities in the ER diagrams include all the data that will be saved in the database in a form of attributes. The ER diagram is then in fact a synoptical model of the database tables, which will be physicaly created in the database.

For the data structure specification of the system being developed there is several possible models and tools. The Entity-Relationship diagrams is well widespread and is very transparent. There exist many notations for the ER diagrams. Also different tools offer their own

notations. One of the most widespread notations is the Crow's foot notation. It is described on the following two pictures:

Figure 16: Crow's foot notation - relationship description

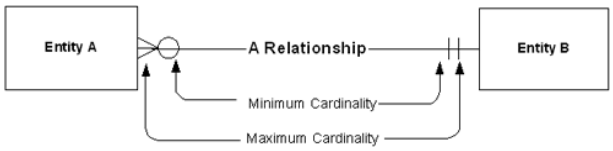
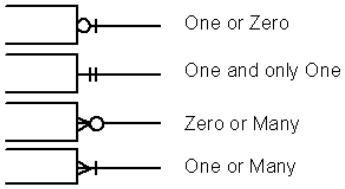
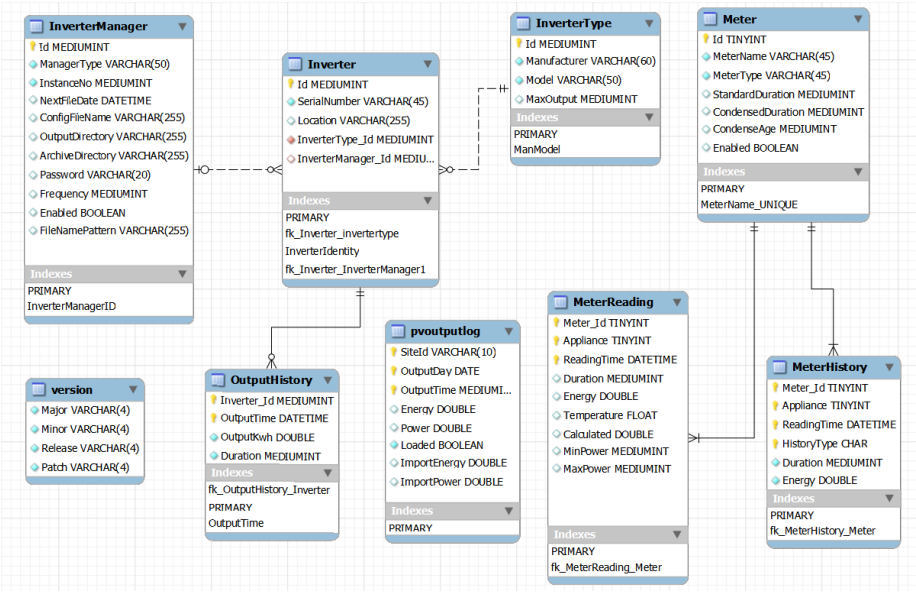


Figure 17: Crow's foot notation – types of relationships



On the following picture then there is an example of the ER diagram in the Crow's foot notation:

Figure 18: ER diagram Crow's foot notation example



Among other possibilities it is possible to use the UML Class diagram, which offers all necessary functionality and has some big advantages, such as the possibility to take the Domain diagram explained before and extend it into the ER model by adding all attributes. The UML

also offers better possibilities of constraint expression via usage of the OCL (Object Constraint Language), which description is out of the scope of this thesis, for more information please navigate to (12).

3.4.1.10 State Diagram

Sometimes software analysts need to depict the state in which different objects, or subsystems can appear within the software solution. A state is a stage of a business data object‘s life cycle. The State diagram shows these states along with the event that initiate the transition of the particular object from one state to another. The most basic elements a State diagram drawing tool needs to provide are the start, end and intermediate state and a transition arrow:

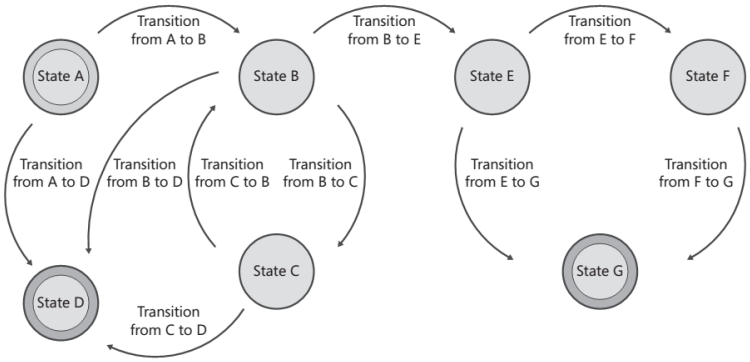
Table 5: State diagram components

Element	Description
	An intermediate state is any of the states an object can be in except the initial and the end state (if those exist).
	The label in the circle is the name of the state. The initial state that a business data object exists in before any of the transition events occur. This element is optional, meaning that none of the possible states an object can be in has to be initial.
	The final state that a business data object exists in. The object cannot be affected by any events after it is in this state. This element is optional, again meaning that none of the

	possible states an object can be in has to be final.
	Visual representation of the initiation and the flow an object undertakes when going from one state to another. The label indicates the event or condition that caused the transition to occur. The arrow is always unidirectional, indicating the direction of the transition.

Source: (10)

Figure 19: A State diagram example (10)



3.5 Business Process management and analysis

There is usually many process diagrams created before a software is developed, describing in detail all processes performed by both people and systems. Best practices suggest that there should be several levels of the depth in process modeling and modern software tools support these levels by making it possible to create "sub processes", which can be opened similarly as a folder.

The process "mapping" is used on such projects where the software analyst needs to understand how some complex business activity works. Then it is necessary to have all the

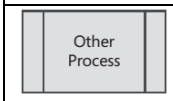
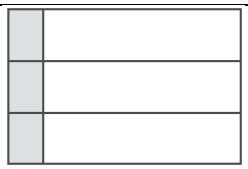
business processes mapped in order to be able to come up with particular functionality of a software system and the data flows within it (data flows diagrams will be described later on). To come up with the future processes without understanding the contemporary ones is very difficult. The Process diagram needs to list all activities in each of the processes, in the same order as they are really carried out. The business processes described are either simply duplicated from those that exists, but in many cases the contemporary processes are altered to suit more the new system and business needs, or completely new processes are created.

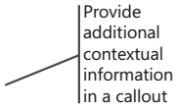
3.5.1 RML Process Flow model

The RML suggests its own notation for process management, which is quite simple but somewhat restricted set of elements:

Table 6: The Process Flow Elements

Element	Meaning
	This is a basic process step that a user takes .It is named with a verb phrase.
	The directional arrow connects process steps or other elements to one another. The direction of the arrow shows the order of the steps .If the line is coming out of a decision step, it is labeled with the decision choice it represents.
	The decision step splits Process Flows in specific ways based on the choices from this step.
	The outgoing references used to show that the flow is moving to another flow.

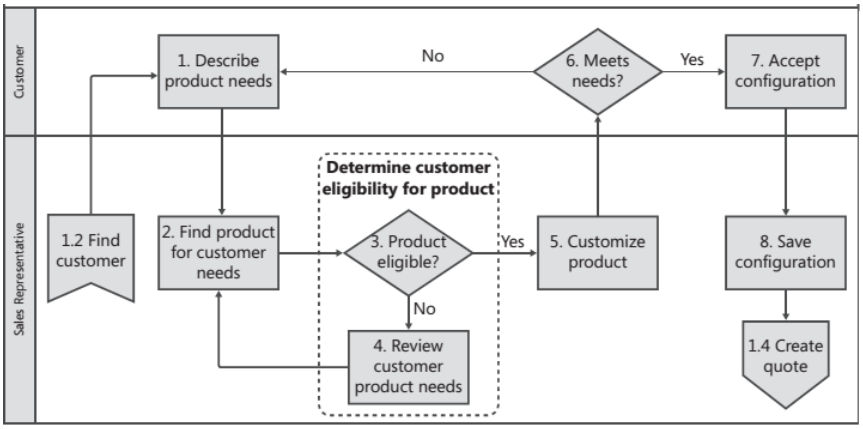
	This is typically used at the end of each lower-level Process Flow to indicate which flow comes next.
	The incoming reference is the complement to the outgoing element. It indicates that the flow is resuming from another flow. This is typically used at the beginning of each lower-level Process Flow to indicate which flow came before.
	The other process element references another process mid-flow and bounces back to this flow after that process ends.
	Swim lanes divide the Process Flow to show which roles are executing the steps. Swim lanes' names are names of users or groups of users or systems engaged in the process.
	The fork and join symbols are identical; the first one in a Process Flow indicates a fork, and the one that follows is a join. A fork splits a process to show that although all steps must execute, they do not have to be done sequentially .The fork allows the Process Flow to avoid forcing sequencing where no sequencing exists. A fork will always be followed by a join at some point further down the flow. All steps before the join must be executed before the next step after the join can be processed. A join will always be preceded by a fork somewhere in the flow.

	Forks and joins can be rotated to any angle when it helps with the layout of the Process Flow.
	A callout is used to provide additional contextual information about a specific activity or event.
	Groupings add additional information to a diagram for readability .Typically, a grouping surrounds a sub process that is not split out into its own flow. The grouping is usually named after that sub process.
	An event indicates that something external to the process happens during the process. An event is part of the normal flow. For example, an event can occur at specific times, after a length of time, or indefinitely until another event occurs.

Source: (10)

Now follows an example of a customer in a shop and a sales representative customizing the demanded product according to the customer's needs:

Figure 20: example of RML Process Flow (10)



While for simple process modeling this notation is sufficient, its capabilities are not quite as wide as might in some cases be necessary. Furthermore, not all the tools support this notation as it is not so widely used. That is a reason why the set of tools from the RML will be here extended for much more comprehensive and widely used notation – the Business Process Management Notation. The thorough definition and explanation of the notation is beyond the scope of this thesis as it has more than 50 elements and the overall complexity of it is vast. However, as the BPMN notation will be used in the practical part of this thesis, the basics will be explained.

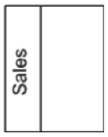
3.5.2 Business Process Management Notation (BPMN)

The full description of the BPMN is at (14) and as it is very extensive (but powerful) tool, only basics will be explained in this paper. The BPMN is a "language" for description of company processes, which appeared in the 2004 as a result of the Business Process Management Initiative and then the further development of it was taken over by the Object management group (15) and now it is at the BPMN 2.0 version.

Among basic elements of the BPMN belong the elements in the following table:

Table 7: Basic elements of the BPMN diagram

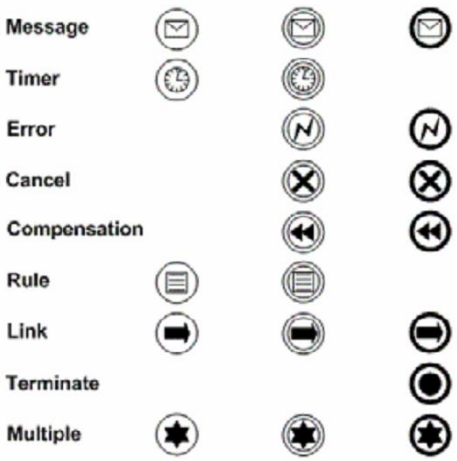
BPMN 2.0 icon	BPMN 2.0 element	Description
	Start event	A start event is an initiative which starts a new process instance
	End event	An end event is the last event of a process after happening which the process is terminated. Note that the end event is distinguished from the start event by thicker line.

	Business event	A business event is an impulse in a business context. There exist many business events defined by BPMN, the icon shown is an “error” event, for complete list of icons please refer to the figure 22 or navigate to (14).
	Pool	A pool represents the container for the activities of a process. Best practice is to use the process name for the pool name.
	Lane	A lane represents a role within a process model. In most cases, this is an organizational unit or a role definition (which is effectively a person or a system taking part in a process).
	Task/Activity	a unit of behavior of a business process element of a business behavior

	Sub process	Sub process is a further divisible part of the main (superior) business process
	Parallel gateway	A parallel gateway is used to indicate that activities can be executed simultaneously or that all incoming activities must be completed before the process progresses to the next activity.
	Exclusive gateway	An exclusive gateway is used for conditional logic. Based on a condition, only one of the outgoing sequence flows will be followed
	Message flow	A message flow is used to send a signal or message from one pool to another. It may not be used to connect activities within one pool.
	Sequence flow	A sequence flow connects activities, gateways, and events to each other within one pool. Therefore, It represents the orchestration of the process definition.

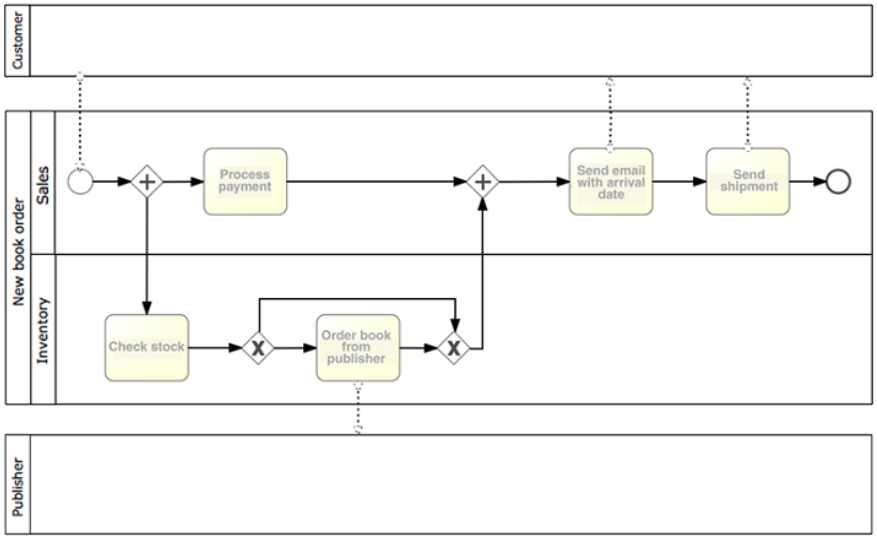
Source: (10)

Figure 21: Types of BPMN events (16)



A business process then consists of activities, tasks, events and other sub processes, so the overall process can create a tree structure of sub processes. Every activity or a task is performed by an actor (a role). On the following picture there is an example of a BPMN process.

Figure 22: BPMN process example



3.6 Project management

Before we dive into project management methodologies it is necessary that we understand what a project is.

The Project Management Institute in its book 'A Guide to Project Management Body of Knowledge (PMBoK Guide)' defines a project as follows:

"A project is a temporary endeavor undertaken to create a unique product, service, or result. The temporary nature of projects indicates a definite beginning and end. The end is reached when the project's objectives have been achieved or when the project is terminated because its objectives will not or cannot be met, or when the need for the project no longer exists. Temporary does not necessarily mean short in duration. Temporary does not generally apply to the product, service, or result created by the project; most projects are undertaken to create a lasting outcome. For example, a project to build a national monument will create a result expected to last centuries. Projects can also have social, economic, and environmental impacts that far outlast projects themselves."

The Prince2 methodic (Managing Successful Projects with PRINCE2, 2009) defines a project as follows:

"A project is a temporary organization that is created for the purpose of delivering one or more business products according to an agreed Business Case."

According to IPMA a project is defined as follows:

"A project is a time and cost constrained operation to realize a set of defined deliverables (the scope to fulfil the project's ibjectives) up to quality standards and requirements."

From the definitions above we can determine the difference between a project and a process. The difference is quite often not fully understood. In a nutshell the difference is that while a process is first defined and thereafter is run continually and is, in theory, never ending, a project is planned at the beginning and run only once – going through various stages of so

called project lifecycle. During a project there can be many processes being performed but these are terminated as soon as the project is done. Most important facets of a project work that distinguish it from a process work are following:

Aim

The objective of a project is always to introduce a change. Change can be a created product (having a new product is a change), changed processes, changed conditions etc.

Temporary

Projects are temporary from their nature. Once the desired outcome is delivered the project is terminated. Yes, there can be a service provided for the product of the project, which however is no longer work on the product creation itself – the service provision is a process. Projects have a determined start and a determined end.

Cross-functional

In most project people from various areas of expertise are needed to put their knowledge together and create the desired outcome – a change that impacts prople outside the team. A project span often crosses the boundaries of a functional division of an organization and not rarely its spans across different organisations.

Unique

One of the greatest things about projects is their uniqueness. There are no two projects that would be the same. There can be a common framework followed, best practice processes can be introduced and projects can be very similar in nature, there will always be many things in which the projects will differ. Starting from a different team, different client, different location,

different product, different problems, priorities, events etc. All those render every project unique.

Uncertainty

All previously mentioned facets mean that there will both treats and opportunities occuring during any project. Projects are risky and with all the activities, occurring threats and opportunities, along with the people with different expertise needing coordination it is crucial to manage every project.

The Project Management Institute in its book 'A Guide to Project Management Body of Knowledge (PMBoK Guide)' defines a project as follows:

"Project management is the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements. Project management is accomplished through the appropriate application and integration of the 42 logically grouped project management processes comprising the 5 Process Groups. These 5 Process Groups are: Initiating, Planning, Executing, Monitoring and Controlling, and Closing."

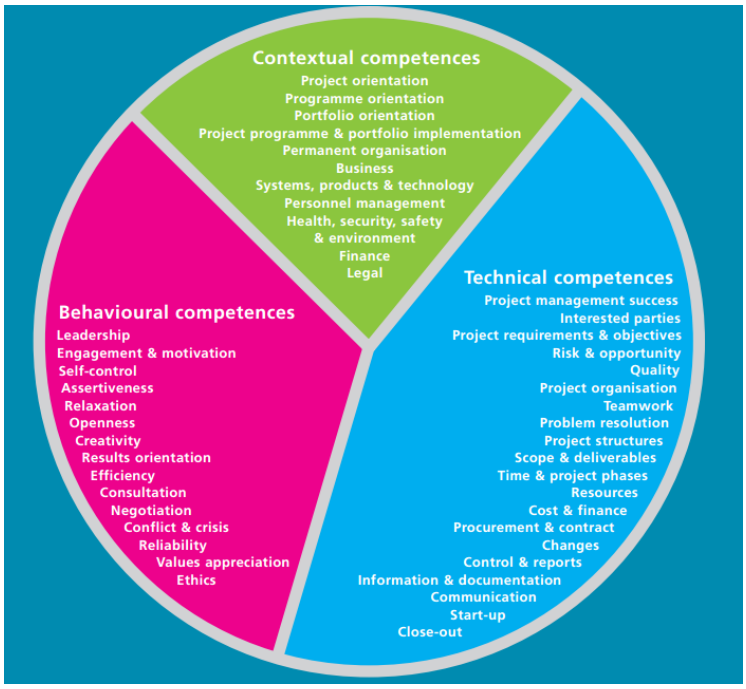
The 42 processes the definition is talking about are defined in the PMBoK Guide, which will be described later on.

The Prince 2 defines the project management as follows (17):

"Project management is the planning, delegating, monitoring and control of all aspects of the project, and the motivation of those involved, to achieve the project objectives within the expected performance targets for time, cost, quality, scope, benefits and risks."

The IPMA standard (18) does no provide a definition of the project management as such. It however offers an overview of the areas a successful project manager has to be competent in which shows how complex the occupation of a project manager is. Those competences are divided into 3 areas – Contextual competences, Technical competences and Behavioral competences:

Figure 23: Competences of project manager (18)



Finally here is a quote of a short phrase from the IPMA book (18), which summarizes everything that has been so far written about project management: “Project management is a fascinating and challenging profession”.

3.6.1 Project Roles

Besides the project manager and a few direct members of the realization team there are also other people who in one way or another take their part in the project. All people being somehow connected to the project as a group are called ‘stakeholders’. The main roles on each project are: Project Manager, Project Sponsor, Project Board, Supply Manager, Project Team Members, Project Administrator or Coordinator, Systems Developer, System Administrator, Programme Manager and Key User. Please find the definition and description of those in the attachment of this paper. It of course does not mean that the list is finite, for specific project might be other roles defined.

3.6.2 Project Triple Constraints

In order to be considered successful every project has to be accomplished with well weighted so called ‘Triple Constraint’ (also called Project Management Triangle or Iron Triangle). The Triple Constraint is a metaphor for a triangle which sides represent the available ‘Time’, ‘Cost’ (in some literature we can find ‘Resources’ instead of Cost) and Scope (in some literature we can find the third one to be ‘Quality’ rather than Scope). The principle is, that these three elements of a project should be held in a certain ‘equilibrium’. It means that effective management of these three dimensions is necessary in order to deliver good quality product, under allocated costs and time. As soon as any of those three dimensions is not held within the client’s expectations and agreed boundaries the project will not be seen as successful by the client even if the other dimensions would be better than expected. During the lifecycle of almost any project a request for change will be raised by the client. It is crucial that the project manager deals with the change by adjusting the three dimension at the same time, steering the project again into the state of ‘equilibrium’.

Figure 24: Project Triple Constraint (19)

In every project there are also supporting processes which ultimately lead to good control of the time, scope and cost. Those are quality management, human resources management, communication management, risk management, change management, supplies management and many more. Each project has its specifics and may require additional processes to be performed during its lifecycle.

3.6.3 Project management methodologies

For managing the whole project, there are methodologies, which purpose is to help a project manager to cope with all processes that need to be managed during the project's lifecycle and provide the project manager with tools, using which he can easily achieve effective and flawless control over those processes.

According to literature and internet resources there seem to be three most widely used methodologies for project management, at least as far as information and communication technology projects are concerned. Those Are PMBoK from the American Project Management Institute, the PRINCE2 from British Axelos and IPMA from the Swiss International Project Management Association.

While these project management frameworks are to some extent interchangeable among themselves and are possible to be adjusted to most environments where project management is needed, there can be circumstances when one of them would be preferable for certain project. In this part of the thesis these three methodologies will be reviewed and compared. The most prominent features which are typical for each of them will be highlighted in order to see what makes each of the frameworks relevant and applicable in what context and what project management environment or case. A detailed structure and content of those frameworks is however not a part of this thesis. For more information the reader can navigate to the references used in the thesis. On the basis of this analysis and comparison one of those will be followed in the practical part of this thesis when making project planning and documentation.

3.6.3.1 Project Management Body of Knowledge (PMBoK Guide)

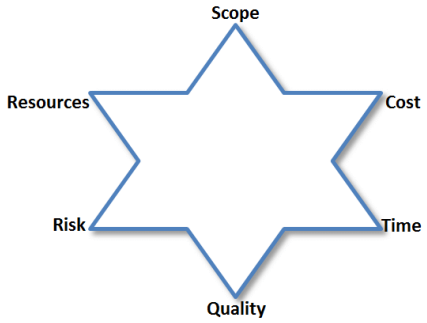
The PMBoK guide from the Project Management Institute (PMI) is a traditional project management framework, which is made to be as general as possible. It is meant to fit as wide spectrum of types of project as possible. Nonetheless its popularity is mainly in civil engineering as it was originaly developled for construction projects. Project managers from other fields (Information Technology for example), however, are also keen to use the framework as well.

The PMI for the purpose of better applicability of the framework developed also field specific extensions such as one for IT (20).

The framework has been growing and with many contributors evolved into todays mature form. The PMBoK is very well suited for projects with many expers from different fields and they all need to be involved in the decision making. In fact, according to this framework, the project team should be participating in task determination, effort estimation as well as schedule creation, identification of risks and their mitigation. This is what differentiates this framework from the other two mostly. It promotes more cooperation of the team on the mentioned activities, while the other two leave these activities mainly on the project manager and executive management.

It is also worth noting, that the PMBoK Guide directly after the Project Management definition describes the dimensions of the project management as they are in the Project Management Triple Constraint, but expands it for the Risks, Resources and Quality dimension to form a sort of star:

Figure 25: PMBoK project management dimensions (21)



In conclusion, the PMBoK is mature and comprehensive framework for project management. It has to be said though that as far as small and medium size projects are concerned (most of IT projects), PMBoK is rather too complex. The excessive bureaucracy might be redundant. In order to stay effective it would need a lot of tailoring and adaptation of the practices of PMBoK and keep only what is necessary.

PMBoK certification

It is possible for project managers and team members to get certified by the PMI to prove their knowledge and competence in the area of project work. PMI offers 3 level certification which take the form of written exam:

1. Certified Associate in Project Management (CAPM)
2. Project Management Professional (PMP)
3. Program Management Professional (PgMP)

Besides those 3 levels, PMI offers also specific certifications aimed at acknowledging competences in particular areas of project management:

- PMI Scheduling Professional (PMI-SP)
- PMI Risk Management Professional (PMI-RMP)
- PMI Agile Certified Practitioner (PMI-ACP)

3.6.3.2 IPMA's ICB

The IPMA is an association of over 55 individual member associations from various countries all over the world. It was established already in 1965 under the INTERNET designation and then renamed to International Project Management Association.

Its standard for project management called the IPMA Competence Baseline (currently in version 3.0) is aimed at building, assessment, and certification of the "competences" of project managers. It is not then a cook book for exact processes and activities conducted during a project, rather the ICB is concerned with knowledge, abilities, experience and qualification of person who will be applying the activities of these processes (the project manager) and what and how he will be performing. Yet, it suggests basic processes for some particular situations. "The ICB doesn't recommend or include specific methodologies, methods, and tools … Methods and tools may be defined by the organization. The project manager should choose the appropriate methods and tools for a particular project situation." (18)

The ICB 3.0 standard is devided into 3 main competence areas:

- Technical (knowledge of methods, techniques and tools)
- Behavioral (soft skills)
- Contextual

These three areas together represent what is called the Eye of Competence. "The Eye of Competencerepresents the integration of all the elements of project management as seen through the eyes of the project manager when evaluating a specific situation.The eye represents clarity and vision. After processing the information received, the competent and responsible professional in project management takes appropriate action." (18)

It's worth mentioning that the IPMA allows the national level member association to adapt the standard to cultural differences by allowing them to adjust the ICB to national versions (NCB's) which are then more relevant to the local environment.

In conclusion, the ICB is most appropriately used when evaluating the project manager's knowledge, abilities, experience and qualification as well as matching the project size and degree of complexity with the project manager's level so as to maximize the project success. The ICB standard is independent of industry and is applicable to almost any kind and size of project.

IPMA Certification

Since, unlike other organisations, the IPMA is aimed at competence rather than process attitude towards project management it can't only rely on written exams of aplicants as competences and abilities of a project manager are not possible to validate via this form. The exams then take also the form of oral exams and validation of experience.

The IPMA uses 4 level certification:

- IPMA Level D – Certified Project Associate

Lowes level, the only one with solely written exam. No experience is required. It is meant for students and project team member who take the role of analyst and workers of particular part of projects.

- IPMA Level C – Certified Project Manager

  Meant for managers of projects of lower complexity.

- IPMA Level B – Certified Senior Project Manager

  Dedicated to managers of project with higher complexity (possibly with several subprojects)

- IPMA Level A – Certified Projects Director

  The highest level, meant for project directors, who lead project programs and portfolios.

3.6.3.3 PRINCE 2

The PRINCE2 standard originally developed by the British government, in particular the Office of Government Commerce, is nowadays owned, further developed and published by the Axelos. It is presented as flexible, process oriented methodology aimed at all types of projects. PRINCE2 was created on the basis of the PRINCE standard from 1989 which was based on the PROMPT methodology which was dedicated to IT projects. Likewise the other standards it is based upon the Best Practice principles, meaning that it is based on mothods and processes proposed by specialists as well as actual users of the methodology.

The PRICNCE2 strongly separates the Project Manager role and the Team Manager role, which makes the standard more suitable and appealing for those who have more technical than people and soft skills needed to lead and manage workers of project directly.

Moreover, the PRINCE2 classifies the individual parts of project lifecycle (called stages in PRINCE2 terms) to Management stages and Technical stages. This enables the project managers without the technical knowledge of the field the project is carried out in to leave and delegate the technical stages to the field professionals and technically more experienced workers or team leaders.

In PRINCE2 the Project Manager merely steers the project, but decisions and plan approvals are to be done by the Project Board. The Project Manager then reports to the Project Board and updates it with ongoing progress. The role of the Project Manager is hence solely to manage daily activities on a project. Of course the Project Board can give the Project Manager some extra authorities, it all depends on particular company and its processes. The Project Board are those who have the power to provide more resources, usually it consists of someone from the client company and superior of the Project Manager (in case Pears Health Cyber this role is called Project Director).

PRINCE2 is business oriented standard, meaning that the existence of a project is continually, from its beginning to its end, justified and its desirability and viability are constantly measured – the Project Board reviews the project and decides and proves if the project is to continue and resources are allocated.

Figure 26: Basic Structure of PRINCE2 (22)

The PRINCE2 methodology includes seven Principles, seven Processes, and seven themes. The seven Principles are mandatory to follow in order for a project to be considered run according to PRINCE2 methodology (17). The following table describes their meaning:

Table 8: PRINCE2 Principles

Principle	Definition
Continued business justification	A PRINCE2 project existence is continually justified
Learn from experience	A PRINCE2 project has defined and agreed roles and responsibilities with an organizational structure that engages the business, user and supplier stakeholder interests
Defined roles and responsibilities	A PRINCE2 project is planned, monitored and controlled on a stage-by-stage basis
Manage by stages	A PRINCE2 project has defined tolerances for each project objective to establish limits of delegated authority
Manage by exception	A PRINCE2 project focuses on the definition and delivery of products, in particular their quality requirements
Focus on products	A PRINCE2 project focuses on the definition and delivery of products, in particular their quality requirements
Tailor to suit the project environment	PRINCE2 is tailored to suit the project's size, environment, complexity, importance, capability and risk

Source: (23)

The Processes are simply put "a set of activities that are required to direct, manage and deliver a project" (23). There are the following 7 processes in PRINCE2:

Table 9: PRINCE2 processes

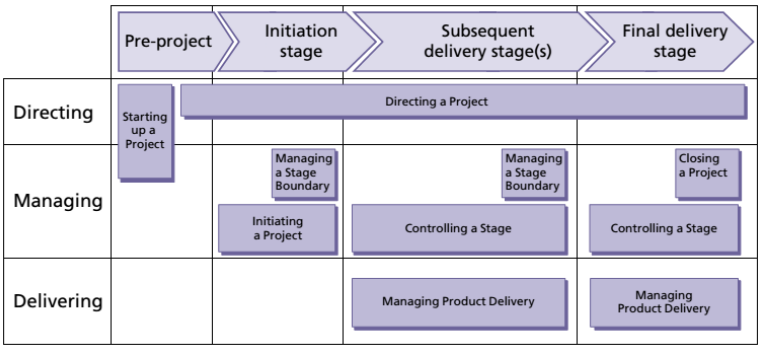
Process	Description
Starting up a Project	Covers the pre-project activities required to commission the project and to gain commitment from corporate or programme management to invest in project initiation by answering the question: 'Do we have a viable and worthwhile project?'
Directing a Project	Describes the Project Board's activities in exercising overall project control. The activities focus on the decision making necessary for Project Board members to fulfil their accountabilities successfully while delegating the day-to-day management of the project to the Project Manager.
Initiating a Project	Describes the activities the Project Manager must lead in order to establish the project on a sound foundation. Every PRINCE2 project has an initiation stage. The key deliverable from this stage is the Project Initiation Documentation, which includes an overall Project Plan and defines baselines for the six project performance targets of time, cost, quality, scope, risk and benefits.
Managing a Stage Boundary	Describes the activities the Project Manager must undertake to provide the Project Board with sufficient information to enable it to review the success of the current stage, approve the next Stage Plan, review the updated Project Plan and confirm continued business justification and acceptability of the risks.

Controlling a Stage	Describes how the Project Manager manages the project execution/delivery activity during a stage, and reports progress and exceptions to the Project Board.
Managing Product Delivery	Addresses the Team Manager's role in supervising the detailed work of creating the project's products and provides the link between the Project Manager and the teams undertaking the project work.
Closing a Project	Describes the closure activity towards the end of the final stage of the project. The Project Manager leads the process which provides for an orderly decommissioning, including any remaining project acceptance and handover requirements.

Source: (23)

The following diagram shows the process model, i.e. where the processes take place during the project lifecycle:

Figure 27: PRINCE2 process model (23)



Finally, the themes are aspects of the project that should be addressed continually (23). PRINCE2 prescribes that Themes should be tailored to fit the project (17). Those Themes are basically a set of guides for how certain activities and processes should be carried out.

PRINCE2 certification

PRINCE2 offeres only two level certification. Both those levels are tested by written exam, following a training. The two levels are:

- PRINCE2 Foundation
    - Lower level
    - Simple written exam
- PRINCE2 Practicioner
    - Higher level
    - Exam based on case studies
    - Condition is to habe

3.6.4 Choosing methodology for case study

In order to create a project plan for the case study in the practical part the most suitable methodology needs to be selected and followed, even though the case study will include only a project initiation and plan.

Having analyzed the three methodologies, the PRINCE2 seems to be most suitable for the case study and IT field overall. It has originally (as PROMPT) been created for IT projects and along with ITIL are methodologies coming from common publishing company, aimed at Information technology management.

Furthermore, as PMBOK is a Standard, not a methodology, it contains a truck load of processes and 'generally accepted' techniques of project management by which to evaluate or complete the way projects are run. It is therefore more theoretical.

IPMA then is rather a tool for evaluation of the level, knowledge and abilities than really a cookbook on how to run projects.

Prince2 on the other hand is a methodology, with a detailed process model and templates. It gives a step by step guidance on how to organise and run a project. It is more

practical than the PMBOK and IPMA. It still needs to be adjusted to needs of particular project environment, but it is more a manual than a reference guide.

In the following preliminary study of the Portfolio management system the planning stage of project management will be carried out. The Prince 2 suggests several techniques in this project stage. The plan should consist of the following:

- ***Product descriptions*** – listing all product descriptions which will be created during the scope of the project plan. The product description in case of software development projects are definitions of the final software. It usually takes the form of a document with requirements and business analysis, which also can (but it's not a rule) be part of the contract.

- ***Project plan*** – which should include the following:

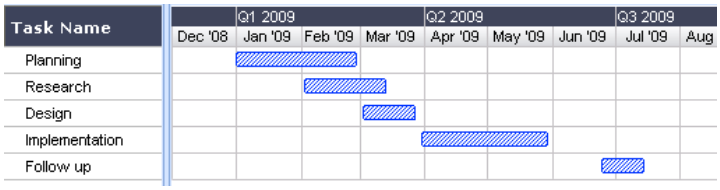
    - **Gantt chart**

The Gantt chart is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is represented by a bar; the position and length of the bar reflects the start date, duration and end date of the activity. This allows you to see at a glance:

- What the various activities are
- When each activity begins and ends
- How long each activity is scheduled to last
- Where activities overlap with other activities, and by how much
- The start and end date of the whole project

To summarize, a Gantt chart shows you what has to be done (the activities) and when (the schedule).

Figure 28: Gannt example (24)



- **Product Breakdown Structure (PBS)**

A project deliverables comprises the top level product that make up the final product, plus all the subproducts that go into creating those top level products. If a top level product doesn't need to be broken down into sub-products, it's called a simple product. The ones that can be broken down are called intermediate products.

The final product is key project deliverable that organizes the team's work into manageable sections. In other words the Product Breakdown Structure visually defines the scope of a project into manageable chunks that a project team can understand, as each level of the Product Breakdown Structure provides further definition and detail.

The project manager creates the Product Breakdown Structure by identifying the major functional deliverables and subdividing those deliverables into smaller and smaller systems and sub-deliverables. These sub-deliverables are further decomposed until a single person can be assigned. At this level, the specific sub- deliverables are identified and grouped together. The lowest level is a list of tasks and "to-dos" to produce the specific unit of work.

- **Product Flow Diagram**

The Product flow diagram is a tool to show in what order the sub-products or features can be built to give incremental increases in capability.This supports one of the core principles in agile development is to deliver a working product as early and often as possible. Into the bargain you have a tangible measure of progress.

- **Activity network**

Activity network diagram of project activities shows the sequential relationships of the activities using arrows and nodes. An activity network diagram tool is necessary for the identification of a project's so called critical path which is such a path in the created diagram where no node (activity) can be prolonged without prolonging the whole project. Therefore it is also used to determine the expected completion time of the project.

- **Resource requirements**

This point is quite clear, once the requirements for the final product are known, the project manager needs to put together which skills he will need for its completion – which human resources he will need, as well as non-human resources if they are needed.

- ***Budget***

When planning a project the budget constrain has to be clear and determined. In software development, the price is usually a part of the constract signed before the actual project start so it is up to the project manager to keep within the given budget.

4 Practical part - Projecting of product portfolio management system

In the practical part of this thesis a preliminary study of a Product Portfolio Management system. During this project a practical usage of a selection of the tools introduced and discussed in the theoretical part will be demonstrated. Besides the demonstration of the tools itself, the objective of this part is accomplishment of comprehensive preliminary study of the system in such a way that key decision makers would be able to make a sensible decision whether or not to make the investment and pursue the project further to actual implementation.

4.1 Preliminary study of Product Portfolio Management system in Pharmaceutical multinational company

4.1.1 Introduction

This project is a case study of analysis, solution proposition and project plan of a Product Portfolio Management software system (PPMS) for a pharmaceutical company, the name of which has to be kept undisclosed as a lot of sensitive information will be revealed during this project. The aim of this project is therefore to analyze the various processes involved in the process of software development by choosing the best process with a goal of describing the importance of system integration.

4.1.2 The client company

4.1.2.1 Description

The client company in this paper is a Swiss multinational pharmaceutical company. The company as a whole achieved net income of 9.2 billion US$, operating income 10.9 billion US$ and 57.9 billion US$ in sales in 2013, last of which makes it number 1 in the world within the pharma industry.

At the top of the company is a holding Group – a Swiss, publicly traded company that owns directly or indirectly its subsidiaries that operate worldwide. The businesses of the holding group are divided into seven operating divisions – original pharmaceuticals, generic pharmaceuticals, eye care, vaccines and diagnostics, OTC products, animal health and drug development. The holding group operates through dozens of subsidiaries in countries around the world, each of which fall under one of the divisions.

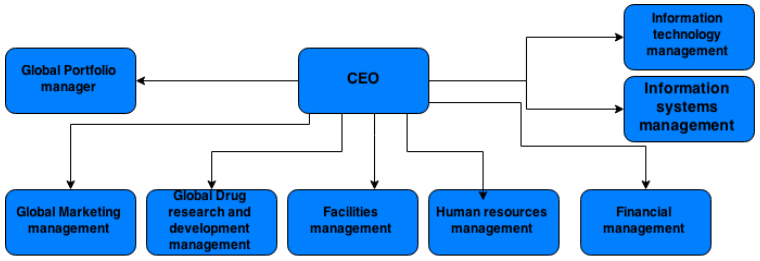
This case study will be prepared in cooperation with one of its subsidiaries – the generic pharmaceuticals, in particular its affiliate based in Prague, Czech Republic. The subsidiary is a leader in the rapidly growing generic industry, which offers more than one thousand different types of high-quality, affordable medicines across a broad range of therapeutic areas. As of 2013, this subsidiary was the second largest generic drug company in the world.

4.1.2.2 Organizational structure

From the PPMS point of view the top level structure is three-fold – there is global management (based in Basel, Switzerland), cluster level management (for example western Europe, eastern Europe, Nordics, Asia, etc.) and country level management.

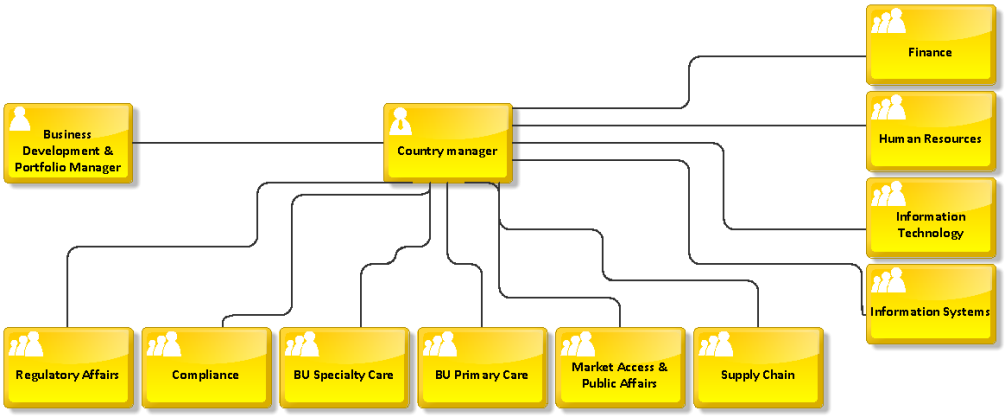
This three level management is important to hold in mind as there will be users of the system to be developed from each of the levels. The high level view on the global management which is situated in Switzerland is as follows:

Figure 29: Org structure HQ (author, 2015)



The organizational structure of the company's typical country level management is shown on the following image:

Figure 30: Org structure country (author, 2015)



Before we delve into the particular field of product portfolio management in pharmaceutical companies, we need to gain an overview of how the company stands on the market and what is its strategy so that all decisions that will be made during the study are in line with it.

4.1.3 The company strategy and business analysis

First we apply the SWOT analysis as the evaluation method so that we can be able to determine the position on the market of the client company for which the system is to be developed. Below is a list of SWOT analysis that were discovered.

4.1.3.1 SWOT analysis

The swot analysis has been put together on basis of talks with consultants from the company, from the company's economic results and from a web site http://www.freeswotanalysis.com (particular page has to remain undisclosed as it includes the name of the company).

Strengths

- barriers of market entry
- high profitability and revenue
- income level is at a constant increase
- skilled workforce
- monetary assistance provided
- good information systems strategy (SAP ERP, Veeva CRM (Salesforce.com))

- Excellent reputation in developing first class Rx pharmaceuticals that maintain high entry barriers to new entrants and competitors
- Wide product range, superb portfolio of young fast growing medicines
- Well developed and rapidly expanding research and development centers
- Second largest maker of generic drugs in the world
- Low threat of new entrants/high competition because of huge investments required to setup a major pharmaceutical firm

Weaknesses

- future profitability
- competitive market
- high loan rates are possible
- tax structure
- future competition
- Long time-to-market
- Declining in R&D productivity
- Growing emergence of generics in the market place
- Brand name harmed due to the legal case against the Indian government
- FDA charged the company with unethical advertising practices
- Lack of resources to exploit prominent opportunities

Opportunities

- global markets (the company can utilize share of costs, risks and production facilities between different countries)
- venture capital
- information systems (SAP, Veeva) can be easily extended by custom applications and systems

- Expanding manufacturing capabilities in developing countries
- Revenue growth opportunity attached to Sandoz and vaccine elements of business will mitigate patent expiry risk associated to Rx portfolio
- Strong research and development base can help in considerable movement to multiple sclerosis and respiratory markets

Threats

- global economy
- price changes
- unexpected problems (economic, political, inside company)
- government regulations
- Major restructuring initiatives
- Uncertain political environment world wide

4.1.3.2 ICT Strategy

As shown in the organizational structure diagram the company has two IT departments the first is operational Information Technology department, which deals with day to day operations and ensures sufficient hardware and networking capabilities of both internal and external IT infrastructure.

The second department is Information systems, its main functions are software solutions, the work of choosing the right software vendors, ensures that information systems is compliant with the business needs and support for internal responsible for gathering internal clients information in terms of information system functionality and applications. This department is also responsible for gathering and managing demands from the internal customers and elevating them in form of proposals for enhancement and future investment in IS infrastructure.

According to their strategy, the company is going to follow the current IT trends in ensuring that the best IT practices are applied in our development. Based on an interview with the Business Effectiveness manager several following points have been pointed out.

Trends in IT the company wants to follow

- Mobile working solutions support an increasing mobile workforce through the use of effective and fit for purpose technologies
- Higher preference for buying technologies from local resellers compared to national resellers, as they tend to be in a better position to provide timely support services and also possess vertical expertise.
- ICT investments are focused on increasing operational efficiencies by c
- Increasingly constrained ICT budgets - looking to deploy technology solutions that offer value for money; therefore, ICT vendors who can offer flexible pricing models will have a competitive edge over their counterparts
- ICT infrastructure is somewhat complex, with several hardware manufacturers, operating systems, databases, applications, and other elements, using multiple instances of different technology types. In means there was no clear strategy in terms of technology which nowadays leads to high costs of maintenance and scattered technology base. The strategy is to unify systems in terms of the technology they are based upon. Mobile infrastructure will be based on the iOS operating system. Servers run Java environments and PHP (apache).
- The company looking to adopt green IT and virtualization, cloud computing, mobility, and business intelligence solutions, as these solutions will help them to reduce their costs, enhance operational efficiencies, achieve sustainability, and improve customer experience.
- Increasing customer satisfaction also has significant influence on the IT investment strategy. The company is looking to utilize collaboration tools such as web and audio conferencing, video conferencing and instant messaging enable knowledge-sharing solutions that can facilitate cross-functional research, core and therapeutic teams to

collect and share data. Also tools such as e-detailing systems, e-mail marketing support systems and virtual rep – systems where a rep can communicate with physicians over the internet are the future of the pharmaceutical business.

- the company considers specific functionality expertise and expertise in the pharmaceutical industry to be the most important criteria in choosing an ICT solution provider
- Managing and monitoring the ever increasing volumes of information across a global organization is a complex task, which is driving the adoption of various enterprise content management solutions.
- In order to curb the impact of competition and regulatory changes pharmaceuticals companies are looking to adopt various predictive analytics and modelling tools in order to make smarter business decisions by understanding the latest market trends and customer preferences.

4.1.3.3 Current product portfolio management in the company

Individual countries maintain Excel sheets, which are updated manually across all the personnel that need to provide some data. For communication, common tools such as e-mails and teleconferences etc. are used. When all the sheets are transferred to the global management, the total number of excel sheets are approximately 1400 lines and 71 columns. There are about 500 strategic decisions done on this basis. Administration and management of (bottom up) forecasts and prioritization. Across all regions, at least 2500 forecasts (or updates) have to be managed per year (requested, received, checked, forwarded, assessed and included into business cases). Done manually on individual Excel sheets.

Launch planning is again in Excel sheets with no connection to the forecasting system and supply chain management system implemented in the SAP ERP system.

There is no central tool and data store available for the Intellectual Property inputs during the process a product creation. There is constantly approximately 1000 active development projects.

The above points bring the risk of missing opportunities in product pipeline and/or investing in products which do not return the expected value. Due to the shortcomings discussed above, the following problems were discovered.

- Non-existing front to end process integration in one IT landscape and across regions
- Lack of visibility on pipeline
- Sub-optimal support of top value drivers “target submission date” and “launch date”
- Enormous data collection and alignment efforts for providing correct and consistent reporting to Top Management
- Difficult to get a single view on the real data
- Fragmented, opaque data sources and tools (Excel mostly) per function / region
- Effective value / risk based decision making requires significant manual effort and may still be in-accurate
- Flexible at area level, inflexible in the aggregate

4.1.4 User Requirements

First we need to describe the basic need of what the company needs. As our company was approached by the client, a very blur and general requirement has been given to us. The Business development and Product portfolio manager of the Czech Republic office said:

“We need a tool that would hold the product data, streamline the portfolio management processes and integrate data from other systems into one common access point.”

Having discussed it further, we gathered that the system will used by offices around the globe and partly by the headquarters. The pilot is to be developed and tested in the Czech office and thereafter spread to all the other offices. Hence the whole analysis and requirements determination will be conducted with cooperation with the Czech office. The main contact person with whom most of the functionality and requirements will be discusses is a Business Development and Portfolio head of the Czech and Slovak office.

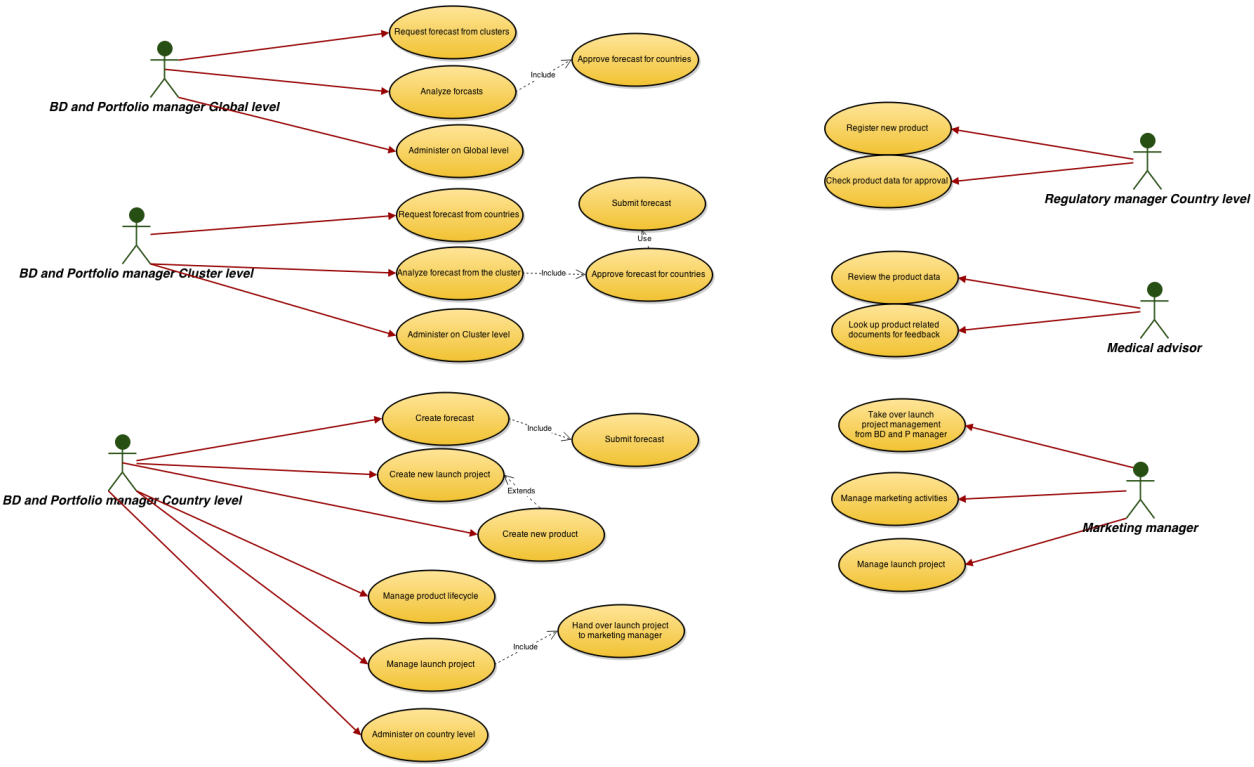
Now we can start gathering actual requirements of what the system should do. The systems will serve mainly for the country level product portfolio management, some input will

be required from the cluster and the headquarters level of management as well. From the initial meetings it seems the system should have 5 types of users, though these will be in many countries throughout the globe. Overall the system should ultimately reach 2000 users. What the system has to enable its main users to do is shown on the following Use Case diagram.

4.1.4.1 Use case diagram

The following diagram depicts users and the high level view of their interaction with the system to be developed.

Figure 31: Use case diagram (author, 2015)



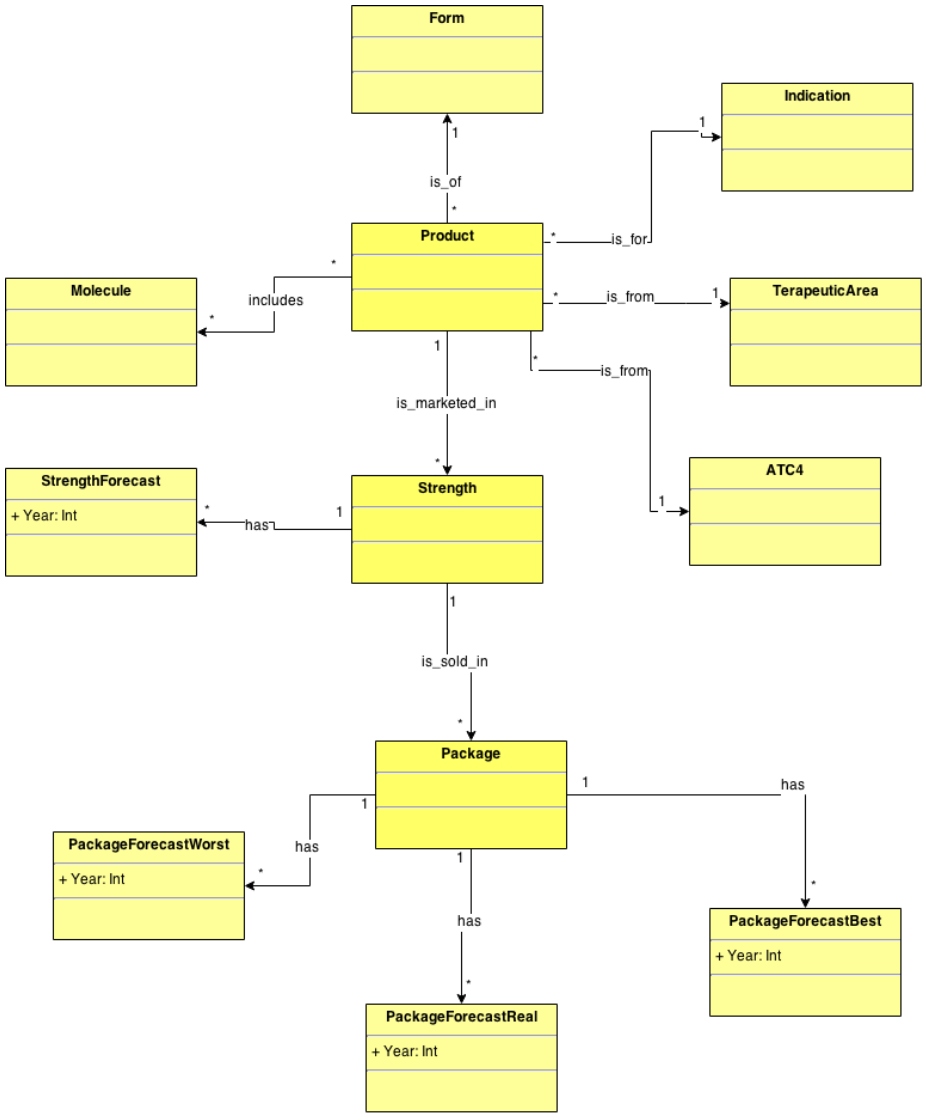
From the Use case diagram we can also see the main users of the system. During process analysis which will come later on in this study, more roles necessary to participate in the processes can be discovered.

The primary role of the system is to hold complete information about the products – drugs of the company. Since every country needs the possibility to manage the products individually, the information needs to be held for each country individually and products will be added by BD and Portfolio managers in individual countries.

4.1.4.2 Product Data:

One of main tasks of the system is to hold information about the products of the company. The following class diagram shows the basic structure of the core part of the system – the Product management:

Figure 32: Class diagram - product data (author, 2015)

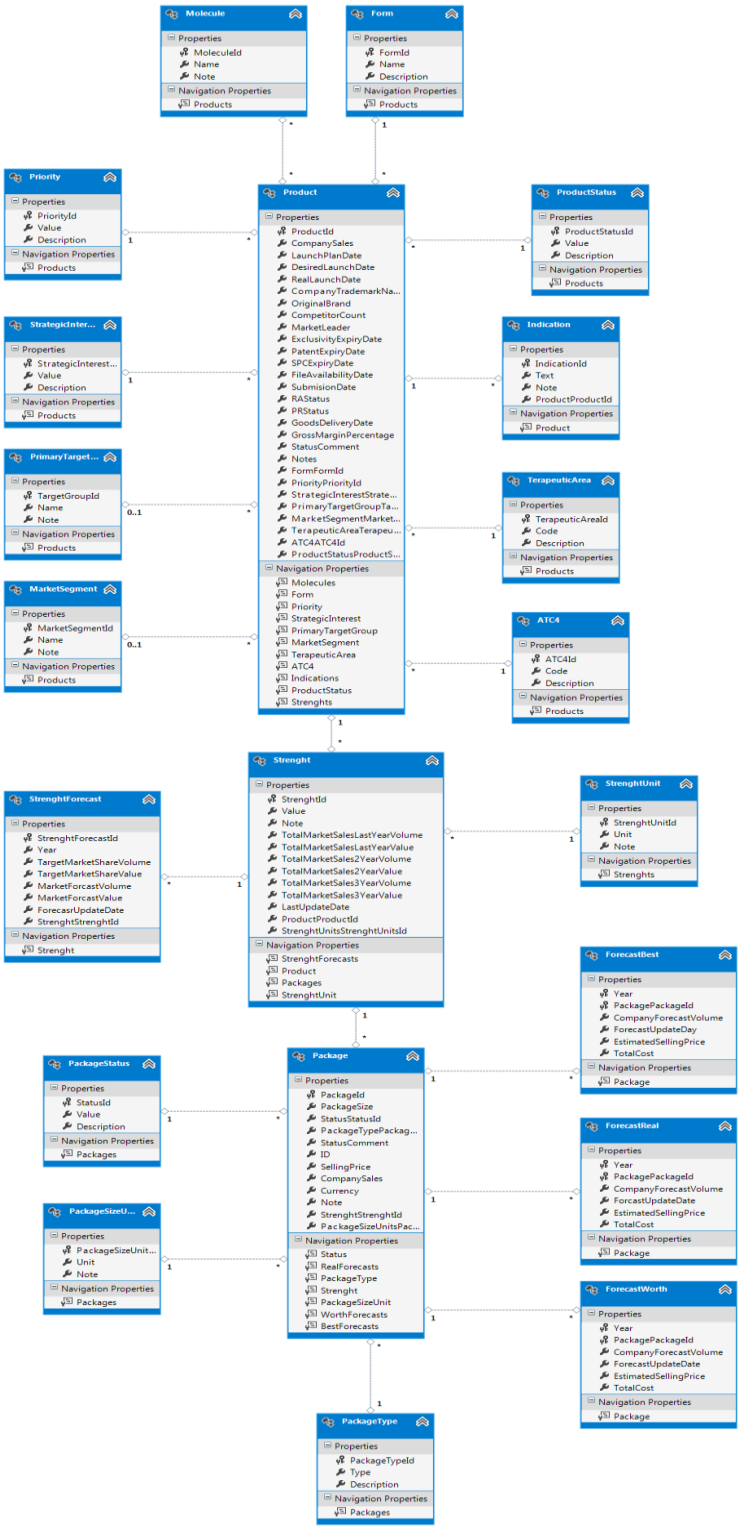


As we can see from the Entity-relationship diagram, there are three main entities – the Product, the Strength and the Package. Product is the class around which the whole system is build. Every product is a medicine aimed at one Indication and belongs into one ATC4 (Anatomical Therapeutic Chemical) group and one Therapeutic Area. The medicine takes a certain Form and consists of one or more molecules. The product then is marketer in various Strengths. These Strengths must be possible to forecast by the BD & Portfolio manager so as the managers have an overview of the expected future evolution of the market. Also this forecast in terms of needed

amounts of each of those Strengths will be used for elevating the forecast to the cluster and headquarters management in order to determine what will be produced by internal resourced of the company or commonly for greater regions. The processes of the forecasting and allocation of the resources will be described later on. Each individual Strength can be sold in different packages. The Packages also need to be forecasted. Since these are the final product which will appear in pharmacies, the precision needs to be very exact. For that purpose the system will enable the BD & Portfolio manager to make best case, the estimated real and the worst case scenario. With these information the manager will be able to make other decisions with more confidence.

The following Entity-Relationship model shows the structure of the data in the context of products in more detail. It includes comprehensive list of attributes the individual objects needs to have for the purposes of the system:

Figure 33: ER diagram - product data (author, 2015)

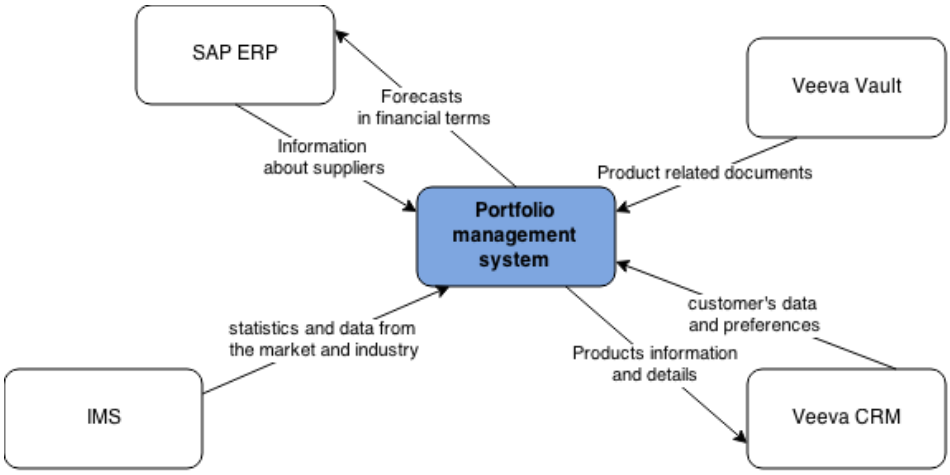


4.1.4.3 Context of the system

The system will cooperate with other system which the company already has in place. Those are the SAP ERP (Enterprise Resource Planning), Veeva Vault – the CMS (Content Management System), the IMS – the Big Data analysis providing system from the pharmaceutical industry and the Veeva CRM (Customer Management System).

The following diagram shows the relation of our new information system to the company's IS infrastructure:

Figure 34: Context diagram (author, 2015)

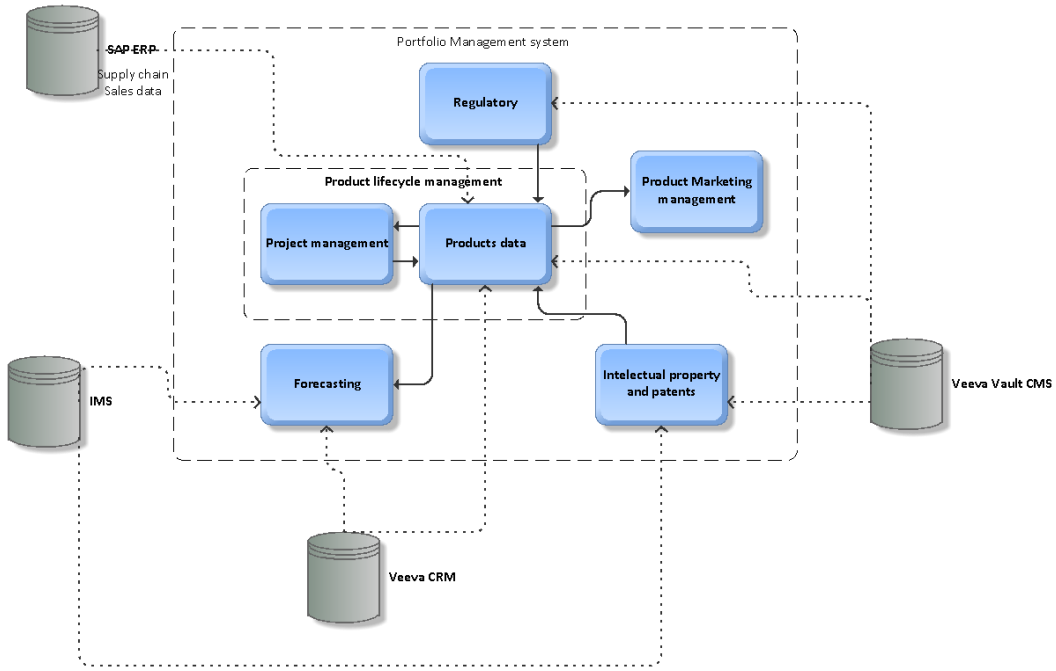


4.1.5 System modules

The system will be used by users of different roles. While the Business Development and Portfolio manager will have access to all functionalities, the other user roles should only have access to certain information the system will provide. To cope with this, the system will consists of modules. Each of the modules will provide view on certain data and certain functionalities. The BD & Portfolio manager will have access to all the modules as well as administration of the application for respective area.

When we take a closer look at the system and take the user requirements into account, we are able to define the structure of the system itself on high level. It will be divided into 5 modules as shown on the following diagram:

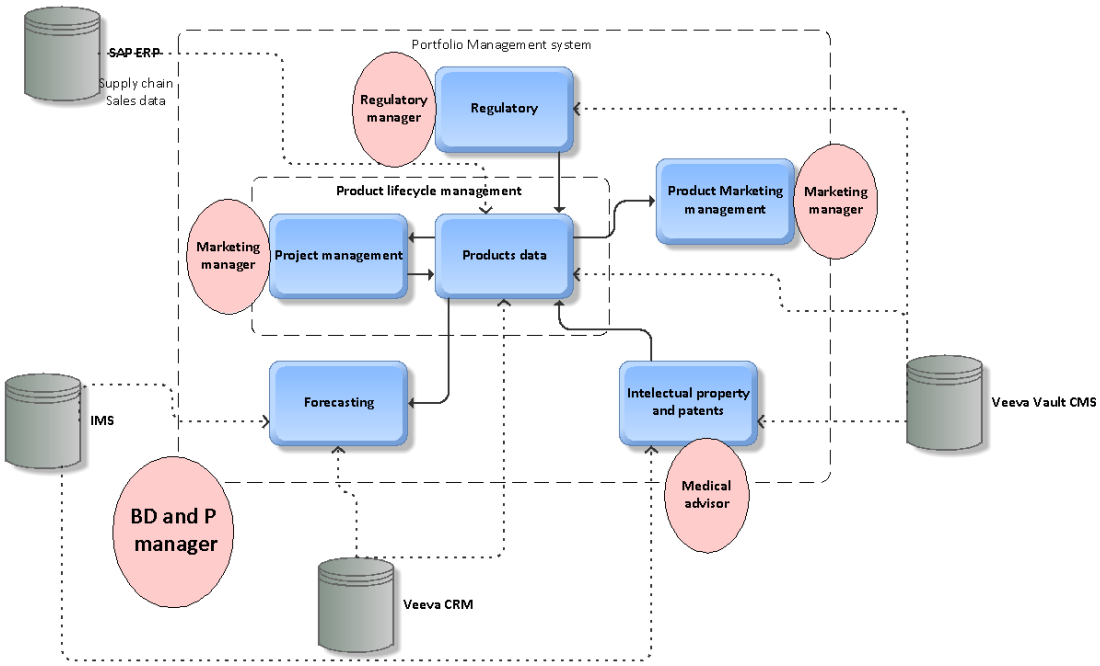
Figure 35: System modules (author, 2015)



Each of the modules will provide users with certain functionalities. Each role of the users should only be able to reach such modules which are deemed necessary for the user's work and participation in the product management processes.

On the following diagram we can see the users of the system and module they will primarily use:

Figure 36: Main users of individual modules (author, 2015)



4.1.5.1 The Product data module

The product data module is the central module of the system. It will hold data with the structure that was described in the Product data chapter. This module will be primarily used by the BD & Portfolio manager himself who will be able to create new Products, their Strengths and Packages.

4.1.5.2 The Project management module

The lifecycle of a product needs to be overviewed and managed as a project. The managers need to maintain a track the time schedule of various activities during lifecycle of a product and keep the timeline under control. For that purpose a classical Project management tool is necessary. It means it needs to provide a possibility to create Gantt diagrams, with scheduling capability and resources assignation to individual activities. This module then needs to notify its users of the deadlines of individual activities via email so they can know what should be provided and when.

This module will be available to the BD & Portfolio manager (all three levels), the marketing manager and the launch manager.

4.1.5.3 Regulatory module

Every product has to be registered in and institute of the respective country. In Czech Republic it is the State institute of Drug control SUKL (25). The Regulatory manager who is in charge of this act need to be able to look up the necessary information from the CMS system (Veeva Vault) review them and then file the registration form with the files required.

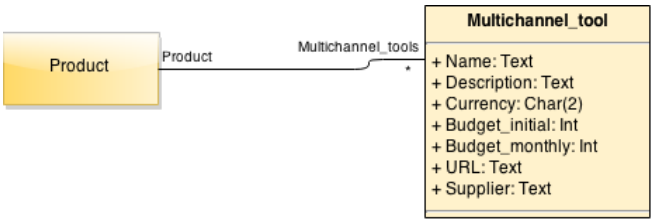
4.1.5.4 Product Marketing management module

Once a product is ready for production, its management is handed over to the Marketing manager whose first task is to lead the process of smooth introduction of the product to the market. It is called the Launch management. The module hence needs to enable the Marketing manager to use different tools of multichannel marketing and measure the efficacy of those tools during the launch as well as during the lifecycle of the product. Among multichannel activities belong for example:

- E-mail
- E-detailing systems (part of Veeva CRM, such as iPad presentations, CoBrowse, Engage and other tools)
- Web sites promoting the product
- Prelaunch Website
- Prelaunch Videos
- Virtual konference with representants

Most of the tools needed for product marketing itself is in the Veeva CRM system. This module will merely serve as information point for the marketing overview of what is used to promote the particular product. In the database there will be a simple relation:

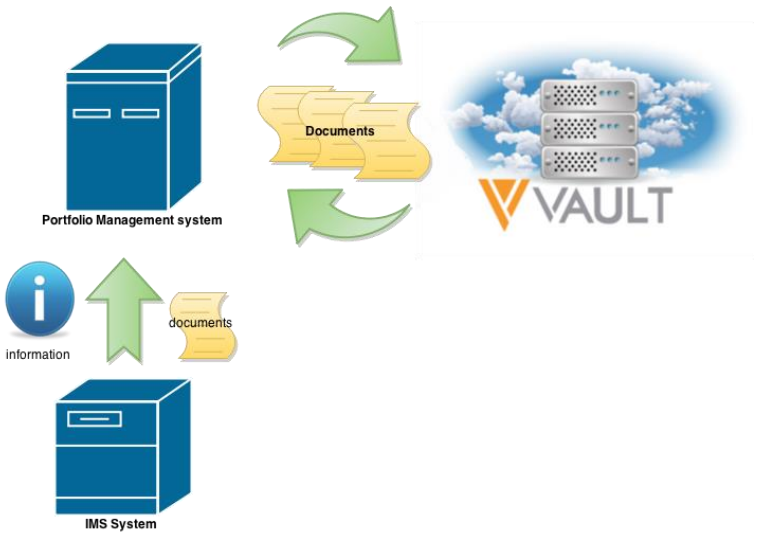
Figure 37: Multichannel_tool class (author, 2015)



4.1.5.5 Intellectual property and patents module

The information about the patents and other documents, clinical studies are easily to be found in systems such as IMS and the Veeva Vault CMS system. However, the managers want to be able to see the important documents regarding a particular product together and be able to search them easily. The system therefore will have this module which will search the documents and information in the IMS and the Veeva Vault systems using their API (Application Programming Interface). It will retrieve the data and enable the user to search in them directly in this system. The system also needs to be able to save a document, clinical study and so on, which is related to certain product. The system will not have its own database for this purpose, it will save all documents into the Veeva Vault CMS system.

Figure 38: System integration for intelectual property management (author, 2015)



4.1.5.6 Forecasting module

One of most important and sophisticated modules in this system will be the Forecasting module – a module that deals with ever challenging problem of efficiently elevate estimated needs of different markets to the higher management and enabling more efficient usage of resources. The central role of the Business Development and Portfolio manager is to forecast and try to estimate the market development in the horizon of up to 10 years ahead and in accordance to that to set the portfolio strategy of the market's company branch so as it fits the future market needs. The job comprises constant analysis various resources, keeping an eye on the market development, reading clinical studies and much more. With accordance to that the BD & Portfolio manager needs to keep his estimation somewhere. As explained before in the "Current Portfolio management" chapter, this has been done so far using excel sheets which were then exchanged by many users after which they reached the headquarters. This system (if we can even call it as such) has proved itself to be utterly insufficient. Hence this new system to be developed will need to support the processes of the forecasting allocation of the resources according to those forecasts.

Forecasting processes:

In order to understand the functionality of the forecasting, which is probably the most important and comprehensive one, we need to create exact process maps. This has been done using the BPMN notation.

The first diagram shows the Forecasting processes, while the second diagram contains process allocation of resources and suppliers.

Figure 39: Forecasting process (author, 2015)

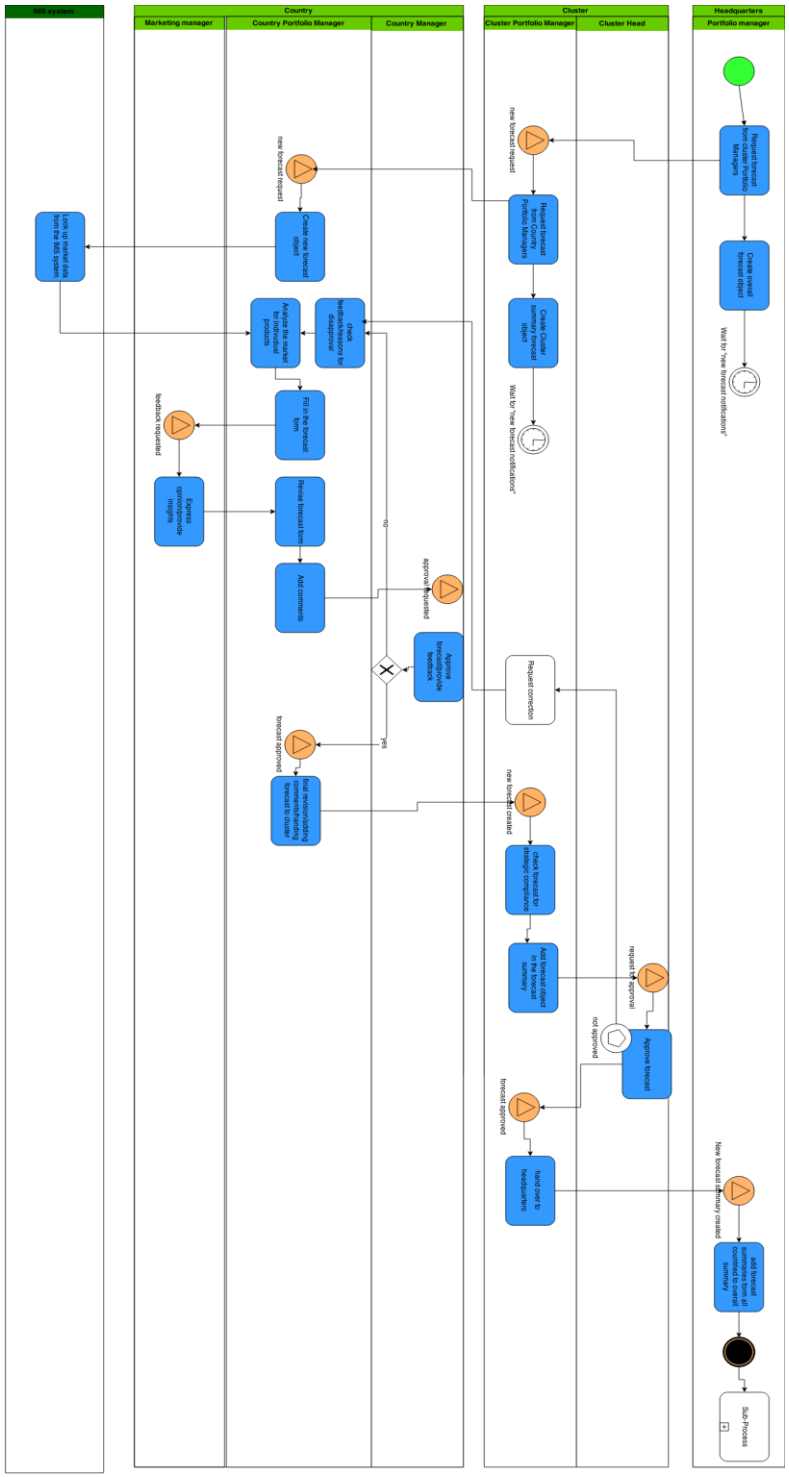
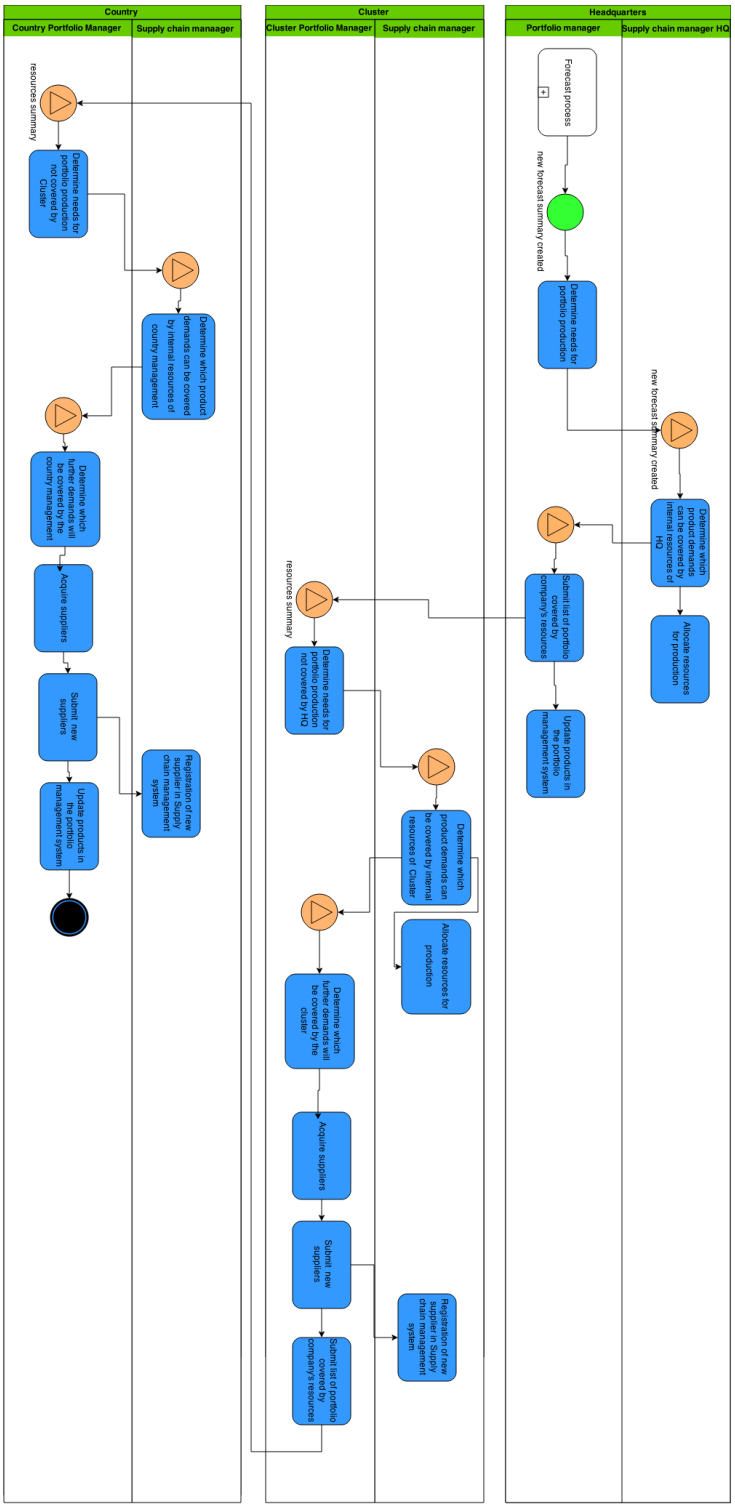


Figure 40: Allocation of resources and suppliers



4.1.5.7 Administration

The system will allow the BD & Portfolio manager administer the system. There will be 3 levels of administration – global which means all items will be available for the whole world, then cluster which means the items will be available to the cluster and then it's the administration for individual countries which items will be available to the certain country.

The administration will be for creating users of the system and then definition of items which will be available in the system as pick list when creating a product. Those include:

- Strength units
- Package size units
- Indications
- ATC4 groups
- Statuses of products
- Molecules
- Forms (of products)
- Strategic interest
- Primary Target
- Market segments
- Package statuses
- Package types

4.1.6 Options for implementation

Due to the current IT trends in line with the Company's strategies described above, the vastness of data and the global nature of the company, the best solution is a Cloud based system. This is very quick and rough estimation of the project, serving as base for choosing the development platform.

4.1.6.1 Technology possibilities

Java platform (spring or J2EE)

- HR prices: 65 Dollars/hour developer, 60 Dollars/hour database specialist, 20 Dollars/hour admin
- More input from Administrators needed + database specialist
- Development time: 200 hour + 15 hours database specialist + 15 hours administrator

Force.com platform

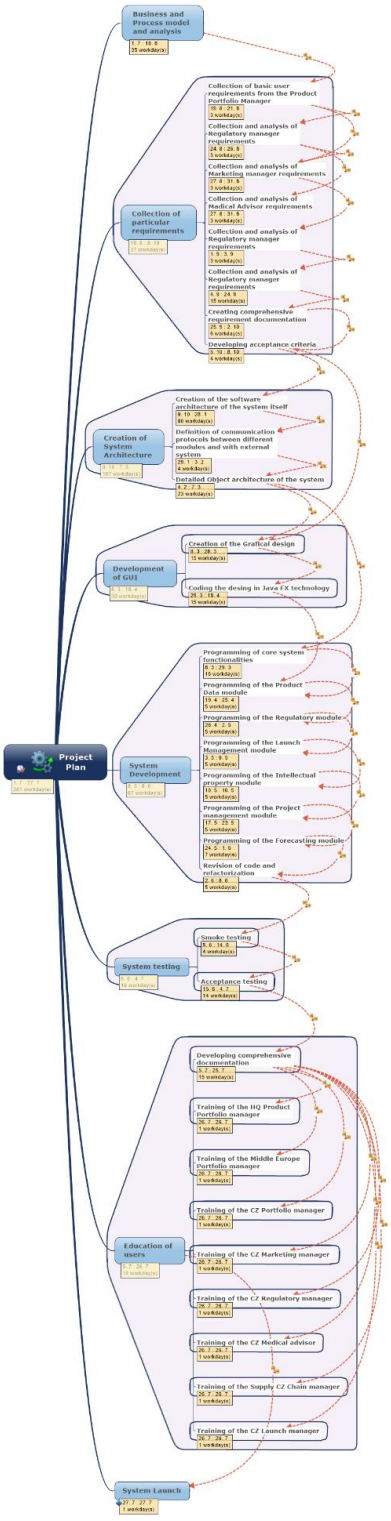
- Developer price: 80 Dollars/hour, 60 Dollars Salesforce administrator
- Development time: 100 hours developer + 50 hours administrator

Having considered advantages and disadvantages of the two platforms, it has been decided that the system will be developed upon the Java platform, as the solution offers above all cheaper implementation of changes during the development phase, which is what is likely to happen.

4.1.7 Project Plan

On the following graph, which was created using a mind mapping tool, an overview of all activities has been depicted that need to be done during implementation of the Product Portfolio management system. This diagram can be also used as so called PBS (Product Breakdown Structure) diagram, which is one of the recommended diagrams to be created when planning a project according to the Prince 2 project framework. The Price 2 also suggests the Product flow diagram, where the sub-products are connected according to their sequential order. This diagram includes that information as well. The project plan only covers the software system development as the company has sufficient hardware to run the system once developed.

Figure 41: PBS and Product flow (author, 2015)



On the image on the next page we can see the Gantt diagram of the project of the implementation of our solution. It includes all the activities that were depicted on the previous mind map image and the dates correspond as well. The gantt diagram is better for visualizing the dependency of individual tasks.

Figure 42: Gannt diagram

4.1.8 Pricing and Human resources

During implementation of the system, we will need certain IT professionals as well as the Key users from the client company to cooperate, provide feedback and set requirements for the system. The table below show the list of roles of IT professionals and users that interact with the system.

Table 10: Needed resources

IT staff	Key users
Project manager	Product portfolio manager country level (Czech Republic)
IT consultant	Product portfolio manager HQ level
Graphic designer	Supply chain manager
Java developer	Regulatory manager
Java architect	Medical advisor
Database specialist	Marketing manager
Administrator	

Source: (author, 2015)

The price of the system will be charged in the prices for the individual human resources. For objective price we need to count on both external costs (cost for actual implementation) as well as internal costs (time of the internal human resources). The list of individual prices and time needed during the implementation project is shown in the following table.

Table 11: Price of individual resources

Role	hours needed	price/hour ($)
IT staff		
Project manager	120	65
IT consultant	40	65
Graphic designer	34	60
Java developer	250	65
Java architect	50	75
Database specialist	20	60
Administrator	30	Internal
Key users		
Product portfolio manager country level (Czech Republic)	30	Internal
Product portfolio manager HQ level	34	Internal
Supply chain manager	5	Internal
Regulatory manager	5	Internal
Medical advisor	5	Internal
Marketing manager	5	Internal

Source: (author, 2015)

The overall system price therefore would be 33 640 $ + 114 hours of internal workers.

5 Results and discussion

The preliminary study has discovered weaknesses in the current processes of the product portfolio management of the client. During the discussions with the representants of the client the most crutial points that a new solution has to solve have been defined.

When coming up with the solution the strategy, prominently the IT strategy has been taken into account so the new solution will be in compliance with it. In particular the solution aimes at streamlining some of the core business processes such as forecasting, launch management, medical review and approval of product related materials and more. The solution will be build on standard company's technologies – Java. To remain complient with the IT strategy which states that systems should be cloud and web based, the system will be web based, hence build with the Java EE technology and other standart web technologies. The solution also integrates data from other systems to provide users with "one point of information" needed for their work.

The solution itself will be divided into 5 modules, which will be activated depending on which role the user of the system has. It is the Product lifecycle management, Forecasting, Intelectual property and patents, Product Marketing management and Regulatory module. Each of those modules then lets its user view the data he needs to see for his role. That includes data from the system itself as well as appropriate data from other systems – SAP ERP, Veeva CRM, IMS Health, Veeva Vault CMS.

Key benefits of the new solution to be developed

- Creation of a harmonised global Development back-bone holding and managing key data from Portfolio Management, Development, Regulatory, Launch supported by Forecasting module, all resulting in optimum Batch supply on optimised economies of scale.

- First to Market: The new system is designed to support the Clients strategy to deliver generic drugs to markets first.
- Reduction in interface management and improved alignment & reporting:
- Currently many duplicate Excel spread sheets and silo databases are running in various departments and functions, e.g. global and US. These are not synchronised and demand huge efforts in reporting, and alignment of data. This will be solved by the new system.
- The new system will be ultimately used by approx. 2000 end users, providing ONE SINGLE DATA REPOSITORY, a source of harmonised & consistent data that can be analysed and used more effectively across functions and regions and which will enable more effective value / risk based decision making along the front to end process (ideation to launch).
- First To File (FTF): improved and optimally integrated processes from portfolio via project management, competitive intelligence and launch management will increase success rates of FTFs and their launch.

An essential part of every preliminary study is, as it servers for the client to decide whether or not to execute the project, the project plan – time schedule and the pricing. The time schedule supposes start of the project 1.7.1015. With that in mind the estimated end would then be 27.7.2016. In order to visualize the time line and the standard Gannt chart has been used and somewhat non-standard mind-map which demonstrates the PBS (Product Breakdown Structure) and the Product flow diagram.

Next steps

The preliminary study serves to the client company as material upon which a decision is made on whether or not to invest resources to pursue the project further. The client knows what he will receive (though the exact functionality is not yet defined) and the approximate price for it.

It will now undergo a decision making process in which the management of the client pharmaceutical company will decide whether to invest into the system. Its members will analyze the value over the resources spent and will make a decision of the approval of spending the resources. The preliminary study serves then not only as a solution proposal but as well as an offer from the supplier to its client.

If the project will be approved then further, detailed analysis of requirements from every role of a user will have to be conducted to design precise functionality of each of the modules. So far the consulting has been conducted only with the BD and Portfolio manager and the Business effectiveness manager.

After the graphics of the system will have to be developed along with the technical architecture, after which the system will be developed and tested. At the end, the training of the users will start.

6 Conclusion

One of the goals of this paper has been to study and review the best practices and tools for capturing and analyzing software requirements and Business analysis. In the first part of this thesis the most useful techniques of the business and requirements analysis have been reviewed based on the study of the literature and my own experience. Even though the list is long it is by no means exhaustive. Only the most used and known techniques and tools were introduced, some of which were later demonstrated in the practical part of the preliminary study.

Another goal mentioned in the thesis assignment has been to review and analyze the most used project management methodologies and determine which one is most suitable for the work on the practical assignment. Second part of the theoretical part of this thesis, the most used Project management methodologies were reviewed and compared in terms of usability in the sphere of Information technology. The outcome of this part was that the Prince 2 methodology seems to be the most suitable one for the IT field and so the Prince 2 has later been used when planning the project of the system development.

In the practical part of this thesis, a case study – preliminary study of a product portfolio management system for a pharmaceutical company has been created. First, the requirements have been collected, formalized and documented using the tools and techniques from the first part of the theoretical part. Then, the project plan has been created and timeline has been scheduled. The plan also includes the resources needed for the implementation of the system and the pricing for the system with respect to the resources needed.

The implementation of the new system should lead to tremendous improvement and achievements of different functionalities which could not be achieved by the former system.

6.1 Key gains of this thesis for the author

During creation of this master thesis the author has learnt many things, namely tools and techniques for gathering, analyzing, formalizing and documenting the requirement for a software system. Furthermore, the author learned about the three most known project methodologies – Prince 2, PMBoK and IPMA. Since from the research it seems that the Prince 2 is most suitable for IT projects, it has been studied a bit more deeply and the author used the knowledge in the practical part when creating a project plan.

In the practical part of this thesis the author had a chance to apply gained knowledge from the theoretical part on a real project – the preliminary study of a product portfolio management system for a pharmaceutical client. The author then selected some of the tools and techniques described in the theoretical part for the requirement analysis and documentation and used them to describe basic functionality of the system. Upon that methods suggested by the Prince 2 methodology were used to plan an eventual implementation project.

7 References

1. **Borysowich, Craig.** Effort Distribution Across the Software Lifecycle. *Toolbox.com.* [Online] October 26, 2005. [Cited: 12 14, 2014.] http://it.toolbox.com/blogs/enterprise-solutions/effort-distribution-across-the-software-lifecycle-6304.

2. **Wiegers, Karl.** *Software Requirements 2.* s.l. : Microsoft Press, 2003. 978-0735618794.

3. **Robertson, Suzanne and James.** *Mastering the Requirements Process: Getting Requirements Right (3rd Edition).* s.l. : Addison-Wesley Professional, 2012. 978-0321815743.

4. **International Institute of Business Analysis.** *A Guide to the Business Analysis Body of Knowledge (BABOK Guide), version 2.0.* s.l. : International Institute of Business Analysis, Toronto, Ontario, Canada, 2009. 978-0-9811292-2-8.

5. **The British Standards Institution.** *ISO/IEC IEEE 24765:2010 Systems and software engineering. Vocabulary.* s.l. : The British Standards Institution, 2010. 978 0 580 62587 9.

6. **Jones, Capers.** *Estimating Software Costs.* s.l. : McGraw-Hill, 1998. 9780079130945.

7. **McConnell, Steve.** *Code Complete: A Practical Handbook of Software Construction, Second Edition.* s.l. : Microsoft Press, 2004. 079-0145196705.

8. **Office of Government Commerce (OGC).** *ITIL: Service Design.* s.l. : The Stationary Office (TSO), 2007. 978 0 11 331047 0.

9. **Wiegers, Karl.** Karl Wiegers Describes 10 Requirements Traps to Avoid. [Online] January 2000. http://www.processimpact.com/articles/reqtraps.html.

10. **Joy Beatty, Anthony Chen.** *Visual Models for Software Requirements.* s.l. : Microsoft Press, 2012. 978-0-7356-6772-3.

11. **Requirements Modeling Language: Visual Models Templates. *seilevel.com.* [Online] [Cited: 12 19, 2014.] http://www.seilevel.com/ba-resources/rml-requirements-visual-models/.**

12. Object Management Group. Unified Modeling Language (UML). [Online] 2014. [Cited: 12 17, 2014.] http://www.uml.org/#UML2.0.

13. [Online] [Cited: 12 11, 2014.] http://twu.seanho.com/11spr/cmpt166/uml/extcasediagram.gif.

14. Object Management Group. BPMN specification. [Online] [Cited: 12 20, 2014.] www.omg.org/spec/BPMN.

15. Island, Shelter. *Activiti in Action: Executable Business Processes in BPMN 2.0.* s.l. : Manning Publications Co., 2012. 9781617290121.

16. NobleProg Limited. *NobleProg Training Materials.* [Online] [Cited: 01 20, 2015.] http://training-course-material.com/images/thumb/b/b6/Element_type_dimension.png/400px-Element_type_dimension.png.

17. Office of Government Commerce (OGC). *Managing Successful Projects with PRINCE2.* s.l. : The Stationery Office (TSO), 2009. 978 0 11 331059 3.

18. International Project Management Association. *ICB - IPMA Competence Baseline, Version 3.0.* s.l. : International Project Management Association, 2006. 0-9553213-0-1.

19. Time, Budget and Scope. How do you manage? *http://www.yourmoneymattersinc.ca.* [Online] [Cited: 12 16, 2014.] http://www.yourmoneymattersinc.ca/2013/02/are-you-an-effective-management-professional/.

20. Project Management Institute. *Software Extension to the PMBOK Guide Fifth Edition.* s.l. : Project Management Institute, 2013. B00I5PE3PU.

21. Mike Iseman, MBA. *Iseman.com.* [Online] [Cited: 12 20, 2014.] http://iseman.com.

22. Prince2 structure. *Verax Consulting.* [Online] [Cited: 12 20, 2014.] http://www.verax-consulting.com/prince2-structure/.

23. Murray, Andy. PRINCE2 in one thousand words. [Online] 2011. [Cited: 12 12, 2014.] https://www.best-management-practice.com/gempdf/PRINCE2_in_One_Thousand_Words.pdf.

24. [Online] [Cited: 12 14, 2014.] http://www.gantt.com/.

25. http://www.sukl.cz/. [Online] [Cited: 01 12, 2015.] http://www.sukl.cz/.

26. Project Management Institute. ***A Guide to the Project Management Body of Knowledge (PMBoK Guide) - Fourth Edition.*** **s.l. : Project Management Institute, Inc., 2008. 978-1-933890-51-7.**

27. Jim Arlow, Ila Neustadt. ***UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design (2nd Edition).*** **s.l. : Addison-Wesley Professional, 2005. 978-0321321275.**

28. Kim Hamilton, Russel Miles. ***Learning UML 2.0.*** **s.l. : O'Reilly, 2006. 978-0-59-600982-3.**

29. [Online] [Cited: 12 12, 2014.] http://www.productbreakdownstructure.com/.

30. [Online] [Cited: 11 19, 2014.] https://www.workbreakdownstructure.com/.

31. ORACLE. Database Design Using Entity-Relationship Diagrams. [Online] July 2001. [Cited: 12 22, 2014.] http://cursuri.cs.pub.ro/~radulescu/bd/er-case/01_00_stud.pdf.

32. Manchester Metropolitan University. [Online] [Cited: 12 14, 2014.] http://www2.mmu.ac.uk/media/mmuacuk/content/documents/bit/Product-Breakdown-Structure-toolkit-v1.pdf.

33. Martin, Robert C. UML Tutorial: Part 1 -- Class Diagrams. ***www.objectmentor.com/.*** **[Online] [Cited: 01 10, 2015.] http://www.objectmentor.com/resources/articles/umlClassDiagrams.pdf.**

34. Lowe, Don. Product flow diagram. ***Don Lowe.*** **[Online] [Cited: 11 15, 2014.] http://www.donlowe.org/project-management/product-based-planning/product-flow-diagram/.**

8 Attachments

8.1 Project roles

In the tables bellow is a list of typical stakeholders partaking on a project. It does not, of course, mean that the list is comprehensive, there might be other roles needed for projects of some specific requirements. The following list has been taken from and is available at the web site of the University of Glasgow, UK (www.gla.ac.uk/media/media_26397_en.pdf). The list was adjusted and extended to fit the Information Technology projects.

Role Title	Project Manager
Role description	
The person responsible for developing, in conjunction with the Project Sponsor, a definition of the project. The Project Manager then ensures that the project is delivered on time, to budget and to the required quality standard (within agreed specifications). He/she ensures the project is effectively resourced and manages relationships with a wide range of groups (including all project contributors). The Project Manager is also responsible for managing the work of consultants, allocating and utilizing resources in an efficient manner and maintaining a co-operative, motivated and successful team.	
Role particular responsibilities	
• Managing and leading the project team. • Recruiting project staff and consultants. • Managing co-ordination of the partners and working groups engaged in project work. • Detailed project planning and control including: • Developing and maintaining a detailed project plan. • Managing project deliverables in line with the project plan. • Recording and managing project issues and escalating where necessary. • Resolving cross-functional issues at project level.	

- Managing project scope and change control and escalating issues where necessary.
- Monitoring project progress and performance.
- Providing status reports to the project sponsor.
- Managing project training within the defined budget.
- Liaises with, and updates progress to, project board/senior management.
- Managing project evaluation and dissemination activities.
- Managing consultancy input within the defined budget.
- Final approval of the design specification.
- Working closely with users to ensure the project meets business needs.
- Definition and management of the User Acceptance Testing programme.
- Identifying user training needs and devising and managing user training programmes.
- Providing regular status reports to the IPSC Programme Board.

Role Title	Project Sponsor
Role description	
The person who commissions others to deliver the project and champions the cause throughout the project. They will normally be a senior member of staff with a relevant area of responsibility that will be affected by the outcome of the project. They are involved from the start of the project, including defining the project in conjunction with the Project Manager. Once the project has been launched they should ensure that it is actively reviewed. The Project Sponsor is usually the one who has to negotiate a path through the tricky diplomatic areas of the project although the Project Manager will most likely be involved in such areas from time to time too!	
Role particular responsibilities	

- Acts as champion of the project.
- Is accountable for the delivery of planned benefits associated with the project.
- Ensures resolution of issues escalated by the Project Manager or the Project Board.
- Sponsors the communications programme; communicates the programme's goals to the organization as a whole.

- Makes key organisation/commercial decisions for the project.
- Assures availability of essential project resources.
- Approves the budget and decides tolerances.
- Leads the Project Board.
- Ultimate authority and responsibility for the project.

Role Title	Project Board
Role description	
This group, normally containing management grade personnel, is responsible for overseeing the progress of the project and reacting to any strategic problems. The group is optional, as the Sponsor-Manager relationship may be seen as the best means of control, but is usually required in large projects that cross-functional boundaries.	
Role particular responsibilities	
• Championing the project and raising awareness at senior level. • Approving strategies, implementation plan, project scope and milestones. • Resolving strategic and policy issues. • Driving and managing change through the organization. • Prioritizing project goals with other ongoing projects. • • Communicating with other key organizational representatives.	

Role Title	Supply Manager
Role description	
The person responsible for managing supplier-side input to the project. Note that this role is usually part of the responsibilities of the project manager, but on bigger projects the role can be taken by another person.	
Role particular responsibilities	
• Ensures that mandatory supplier requirements are met. • Manages the production and approval of the supplier side of the budget.	

- Makes effective use of supplier resources within the approved budget.
- Tracks performance of consultants and takes appropriate action.
- Proactively develops a collaborative relationship with the organization to Project
- Steering Board level.
- Ensures that there are clear communication paths within the project team and the organization and supplier.
- Acts as main point of contact between the supplier and the organization.
- Produces and monitors financial reports including entry and maintenance of all actual time and expense against the master plan.
- Day to day management of supplier staff assigned to the project.
- Quality Assures the work of supplier staff assigned to the project.
- Encourages the transfer of product knowledge and skills to the appropriate staff within the organization.

Role Title	Project Team Members
Role description	
The staff who actively work on the project, at some stage, during the lifetime of the project. Some may have a specific role – for example, the typical software development project Team might include programmers (but might not in case the solution is a business software such as SAP, or Salesforce.com) or the requirements and business analysts.	
Role particular responsibilities	
Team member roles will vary depending on the type of project. Typically they might be to: • Provide functional expertise in an administrative process • Work with users to ensure the project meets business needs • Documentation and analysis of current and future processes/systems • Identification and mapping of information needs • Defining requirements for reporting and interfacing • User training	

Role Title	Project Administrator or Coordinator
Role description	
Responsible for maintenance of the project plan, maintenance and updating of a project website (if appropriate). Provides administrative support to the Project Manager. This role is most likely to be required to be taken by separate person from the Project Manager in larger cross-functional projects, otherwise it is usually also taken by the Project Manager himself as in case of the Supply Manager.	
Role particular responsibilities	
• Sets up and manages support functions covering planning, tracking, reporting, quality management and internal communication. • Produces consolidated reporting to the Project Board, including milestone summary, key issues, risks, benefits, summary of costs incurred. • Establishes standards, tools and procedures for use on the project, including Issue, • Risk, Change and Information Management. • Manages the Project Library. • Reviews project activities for compliance with procedures and standards. • Manages the support and provision of project tools and equipment. • Manages data security, software and license control. • Assists with the production of user documentation. • Assists with testing.	

Role Title	Systems Developer
Role description	
Works with the Project Manager on defining and executing development requirements.	
Role particular responsibilities	
• Working with the Project Manager on definition of development requirements and priorities. • Data Migration. • Interfaces with other systems.	

- Reporting configuration and deployment.
- Set up and maintenance of security rights and access permissions.
- Contributing to technical strategy, policy and procedure.
- Development and operation of technical testing programmes.
- Production of technical documentation to agreed quality standards.
- Reporting on progress/issues to management and users.

Role Title	System Administrator
Role description	
Management and support of the IT system environments	
Role particular responsibilities	

- Management and support of the various environments.
- Network operating systems management and support.
- Database management and support.
- Back-up and disaster recovery measures.
- Contributing to technical strategy, policy and procedure.
- Development and operation of technical testing programmes.
- Production of technical documentation to agreed quality standards.

Role Title	Programme Manager
Role description	
This role is relevant if there are several related projects, in which case the Programme Manager is leading the projects himself, has subordinate Project Managers working on the individual projects.	
Role particular responsibilities	

- Overall management and co-ordination of the programme of projects.
- Contributing to strategy, policy and procedure.
- Management of supplier/contractual relationships.

- Budgetary control of the programme of projects.
- Monitoring of, and responding to, issues at the programme level.
- Providing regular status reports to the IPSC Programme Board.

Role Title	Key User
Role description	
The key users are responsible for the functionality of the system. These are those who are mostly consulted with during the requirements analysis and usually represent the most important users of the system. The practice is that from each user group one is picked as the consultant and Key User who will provide analyst with information on the system functionality.	
Role particular responsibilities	
• Set requirements for functionality • Give feedback on outcomes from analysis and development • Provide consultation on the user specific domain • Provide necessary materials for the required functionality (For example pictures, texts etc. In case he can't provide these material, the Key user should provide contact on who can, who has the authority, access etc.)	