Politechnika Wrocławska

Wydział Informatyki i Zarządzania
kierunek studiów: informatyka
specjalność: inżynieria oprogramowania

Praca dyplomowa - magisterska

SPEM/UML Extension to specification of IS management processes

Krzysztof Kołcz

słowa kluczowe:
SPEM/UML profile
project management process

krótkie streszczenie:

Promotor:	Iwona Dubielewicz	....	....
	imię i nazwisko	*ocena*	*podpis*

pieczątka Instytutu, w którym student wykonywał pracę

Wrocław 2006

Contents

1 Problem presentation

1.1 Introduction

Processes have great influence on today's software development organizations. Company managers are aware, that having good processes in place can significantly increase organizations captivity. This concern not only software development processes. Having proper management processes is also very important, if not more important then having appropriate development processes.

Process engineering area these days is very heterogeneous. Processes can be described in many ways. Description of processes can be formal or informal and can have form of text or other forms. In such circumstances SPEM profile could gain importance as standard process modeling language. Its advantage is that it base on UML language, which is widespread within software development community. It is formal modeling language, which offer graphical presentation of processes, and thanks this it could ensure better and easier understanding of complicated aspects of processes, which described in form of text is not always easy to catch.

1.2 The goals of the work

SPEM Language was designed to model software development processes. Its authors states, that it is very general language designed to not narrow down its use, so it can lacks some elements needed to model more specifics areas like project management. In this work I would like to answer for several questions that arise while trying to model project management processes by using SPME language:

- Can be project management processes modeled using SPEM profile?
- How much SPEM is fitted for project management process modeling?
- Does SPEM need to be extended to be able to properly model project management?

In this work I would like to examine SPEM capabilities and find SPEM lacks. In case of finding such lacks I would like to propose SPEM extensions, which would help to eliminate them, and present, how this extensions could be applied.

In order to do this, I would like to present SPEM elements, and how several aspects of processes can be presented by using these elements

To present how project management processes can be modeled in SPEM language I would need to considered several aspects of projects, and several types of project models. I would like to analyze PMBOK (Project Management Body of Knowledge), which is recognized and probably the best source of knowledge in project management discipline. It contains description of management processes, although it doesn't constitute methodology or framework. To understand how project management processes described in PMBOK can be used, I would like to analyze several software project development methodologies and frameworks, and find out, which management activities are used by them, in which place of project life cycle they occur, and how are their connections to other management and development activities.

1.3 Terms and definitions

In order to go further I would like to introduce several terms and definitions, which will be used in later part of my work.

Project Life Cycle – Project life cycle can be defined as "The complete set of time periods through which a project passes sequentially in a logical and orderly manner"[1]
“Project managers and organization can divide project into phases to provide better management control with appropriate links to the ongoing operations of the performing organization. Collectively these phases are known as the project life cycle.” [2]
“The project life cycle defines the phases that connect the beginning of the project to its end.” [2]
“Project life cycle generally defines:

- What technical work to do in which phase
- When the deliverables are to be generated in each phase and how each deliverable is reviewed, verified and validated
- Who is involved in each phase
- How to control and approve each phase”[2]

Life cycle model is defined as “A framework containing the processes, activities, and tasks involved in the development, operation, and maintenance of a software product, spanning the life of the system from the definition of its requirements to the termination of its use.” [4]

Project - "A process or undertaking that encompasses an entire set of activities having [...] well defined objectives"[1]

Process - "A set of partially ordered steps intended to reach a goal" [1]

Methodology - "A process that documents a series of steps and procedures to bring about the successful completion of a project." [1]
"A methodology is a set of guidelines or principles that can be tailored and applied to a specific situation. In a project environment, these guidelines might be a list of things to do. A methodology could also be a specific approach, templates, forms, and even checklists used over the project life cycle." [1]
"A formal project methodology should lead the work of all team members throughout the life cycle of a project but it may be useful to think about what a project management methodology is not: a quick fix; a silver bullet; a temporary solution; or a cookbook approach for project success." [1]

Framework - "The framework has always meant the various segments of the project and the development methodology are the means of getting from segment to segment." [1]

ISO/IEC 12207 is characterized as “International Standard” that “establishes a common framework for software life cycle processes”. [4]

1.3.1 Terms and definitions discussion

During my researches I found that four terms: software development methodology, software development method, software development model, software development life cycle and software development framework means almost the same. During ma researches I also very often found, that terms "Project life cycle" (or "Project life span") and "Methodology" were very often used as synonyms, for example in book [1].

In my work I'll refer to Project Life Cycle as sequence of phases, defining what technical work to do, and defining the deliverables to be generated in each phase.

As a framework I'll understand segments or components describing processes, which can be tailored and used to create end-to-end software development process. Example of such framework is ISO/IEC 12207 standard, which defines set of processes. This set can be tailored, and process engineer can choose only this processes that are needed. Process engineer can also impose life cycle on chosen processes.

In my work I'll understand methodology as set of guidelines, practices and processes with usually imposed life cycle. RUP, XP, Scrum are examples of methodologies.

1.4 Introduction to process engineering

SPEM is a language that is designed for describing software development processes and as such is part of process engineering discipline. In this chapter I would like to present how SPEM relates to process engineering terms and definitions.

Very good introduction to process engineering discipline is article "A comprehensive view of software engineering" by C. Rolland. In this article Rolland presents four-world framework, which clearly describe context of process engineering discipline:

- System World – "which includes specification at different levels of detail of what the system does" [5]
- Subject World – "contains knowledge of the domain about which the proposed IS has to provide information" [5]
- Usage World – "describes the organizational environment of the information system" [5]
- Development World – "describes engineering process itself." [5]

Most interesting, in context of my work, is the System World, which "includes specifications at different levels of detail of what the system does. It holds the modeled entities, events, processes, etc. of the subject world as well as mapping onto design specifications and implementations". System world in other word contain process modeled in some language, in my case in SPEM. According to C. Rolland, there are several characteristics that describe System World (and what results, process models). These characteristics are called *facets*. There are four facets that characterize System world:

1. Abstraction – abstraction facet distinguish two abstractions of processes – process model and process metamodel. (Abstraction facet can have values of type and metatype). Process of the same type can be characterized by one process model. On the other way, there can be several instantiations of process model. Process model can prescribe how things must/should/could be done, and process is what really happens. A process metamodel is description of process model, so process models are instantiations of processes metamodel. In terms of SPEM, SPEM metamodel is process metamodel, and process modeled in SPEM is process model. Instantiation of process modeled in SPEM is process.

2. Content – content is described in terms of *coverage* and *granularity*. The idea of *converage* can be defined as follow: "according to Dowson, process models can be classified into three groups of models:
   - activity-oriented,
   - product-oriented, and
   - decision-oriented.
   - Since this classification was made, a new group called the contextual model has also emerged."

   Process groups are listed in sequence of complexity. Activity-oriented process models allow only to present activity flows, and don't consider loops or iterations. Waterfall model treated as sequence of activities is example of this model. Product-oriented models allow connecting activities with products, which are outputs from these activities. Decision-oriented process models apart from presenting activity and object flow allow to:
   - "guide the decision making process
   - help in reasoning about the rationale behind decision
   - support the deliberation underlying the decision process itself
   - keep trace of the happenings of a process and their rationale" [5]

   Contextual model, apart from possibility to present Decision-oriented models, allow connecting the decision with the context in which process is.

   Models that are created with using SPEM language allow creating the most complex, contextual models. They allow to present activity-oriented processes through activity diagram, product-oriented processes through ActivityParametr, which connect activity with output and input WorkProduct, and also through using of activity diagrams, where WorkProducts can be presented as inputs and outputs to activities, it allow for presenting decision-oriented processes through using decision points on activity diagrams, and allow for presenting contextual through using object state, which allow to present WorkProduct objects in several states.

   The *granularity* attribute defines how detailed can be process model. SPEM allows for wide range of granularity. The most general is Lifecycle element, which symbolizes the whole software development life cycle. Phases, WorkDefinitions, Activities and Steps are elements that allow to present process models in growing level of detail.
3. Description – "documentation facet is concerned with the form of the process representation". It has two attributes – *form* and *notation*. While Rolland in his article list script, program and hypertext values for form attribute, SPEM is not assigned to one of them. It is graphical modeling language. Notation defines formality level of model. It can have three values – informal, semi-formal and formal. SPEM is formal language.
4. Modularization – modularization is possibility of dividing processes or products into fragments. In SPEM exists such possibility. Processes can be divided also can be build from parts of other processes. There exists element called ProcessComponent which allow for such division. "Process modules can be looked upon according to two other perspectives: abstraction and aggregation" [5] as further described, "[6] proposes two kinds of aggregated modules called route map and fragments respectively. A route map refers to strategies such as delivery strategies, developmental strategies, realization strategies etc., activities and products concerning system development as well as project management. The fragment is a coherent part of a process for system development or project management. Fragments may be linked to a route map which may establish a complete project approach." SPEM fulfill requirements for

aggregation. According to above description, fragment respond to SPEM ProcessComponent, and route map respond to SPEM discipline.
Abstraction is actually not formally supported by SPEM. “Abstraction is used to capture generic laws governing the construction of different but similar process modules. Generic process modules can take form of *framework* or *pattern*. A framework model commonality between different process models, but for the same type of application. A pattern models a common behavior in process construction. It is generic in the sense that is used every time a process model is constructed.” It is possible to create in SPEM process patterns or frameworks, but no in formal way. There is no association or dependency that would indicate that certain process is instance of certain pattern. Process engineer can informally stated, that certain model is using certain pattern.

It is also important for what aim process models are used (the Usage World), and how strict are obeyed. According to Rolland, there are three aims of process models:

1. Descriptive purpose takes real process and describe from external point of view, what happened during process.
2. Prescriptive purpose “lays down rules, guidelines, and behavior patterns which, if followed, would lead to the desired process performance.” Application of prescriptive purpose can range from strict enforcement to flexible guidance. “In the strict enforcement the performance of the process must follow the prescription whereas in the flexible guidance the prescription is such that it can accommodate a large number of ways in which the process can proceed.”
3. Explanatory purpose is used in case of need of choice from several processes on rational arguments. Process model is in such case are used as rationale for such decision.

Very important in my work is also the Development World. This view presents how process models can be build depending on their purpose, and other attributes. Development World and Usage World views are also important in case of analyzing processes or process frameworks such as processes described in PMBOK and ISO/IEC 12207. As in my work I should present how project management processes are modeled in SPEM language it is important to know what kind of process models I’ll be modeling (descriptive, prescriptive, explanatory).

In my work I’ll concentrate on prescriptive purpose. I won’t model existing processes – I don’t have access to description of processes that take place in real companies. I’ll concentrate on prescriptive purpose, and I’ll try to create project management processes on the base of existing methodologies, frameworks and other descriptions. These models could be further used as guidance in software development process.

To summarize, processes can be modeled in many different ways, depending on what purpose they serve. Process models can be presented as modules from which end-to-end processes can be build or as one monolithic process. Process models can be in form of flexible guidance or strict enforcement. They can also be presented at many levels of abstraction or detail. All this facts will be seen in further part of my work.

1.5 Introduction to SPEM modeling

SPEM is a process metamodel. It allows for describing processes in terms of work to be done in process, roles that should perform the work and deliverables that are produced and consumed by process.
SPEM also allow for describing life cycles of projects. It contains such elements as:
Process, which describe end-to-end process of software development, LifeCycle, which describes life cycle of end-to-end process in terms of phases that it contains, Phases which determine frames for LifeCycle in terms of work that is done within phases and deliverables that are goals of phases.
More detailed description of SPEM elements is presented bellow.

SPEM can be considered both, as metamodel, and as UML profile. As UML tools are widespread these days, and these tools allow for extending UML in form of UML profiles, SPEM is more widespread as UML profile. Actually I didn't find tool that would implement SPEM metamodel.

1.5.1 Conceptual Model

"At the core of the Software Process Engineering Metamodel (SPEM) is the idea that a Software development process is collaboration between abstract active entities called process roles that perform operations called activities on concrete, tangible entities called work products. Multiple roles interact or collaborate by exchanging work products and triggering the execution, or enactment, of certain activities. The overall goal of a process is to bring a set of work products to a well-defined state." [7]

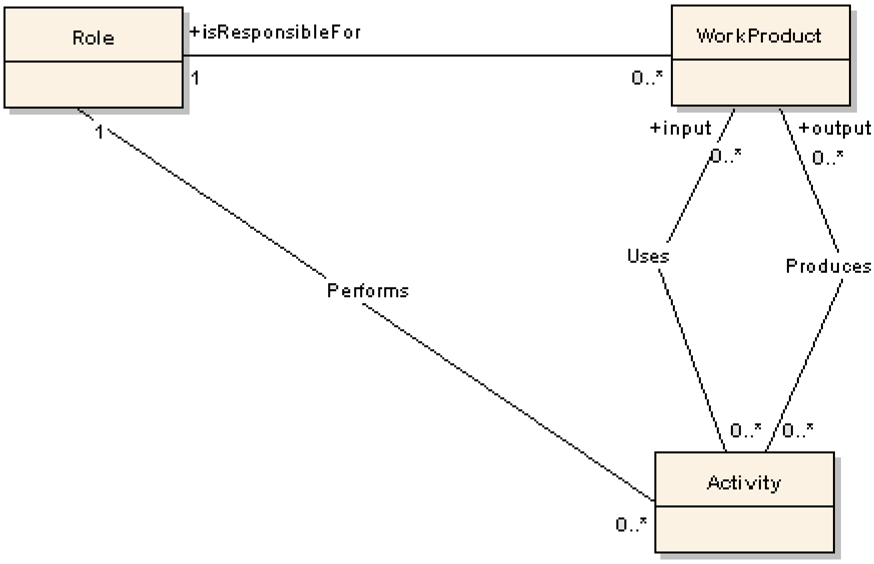


Fig. 1-1 SPEM Conceptual Model

Fig. 5-1 presents contextual model of SPEM. It is not part of SPEM metamodel. It presents only conception of SPEM.

1.5.2 Selected SPEM elements

This chapter contains description of SPEM elements that I'll use in my work based on SPEM specification. Elements are grouped according to packages to which they belong to.

1.5.2.1 Process Structure

Process Structure package contains description of elements that are used to describe processes as presented in conceptual model.

WorkProduct

WorkProduct - A work product or artifact is anything produced, consumed, or modified by a process. It may be a piece of information, a document, a model, source code, and so on. A WorkProduct describes one class of work product produced in a process. Can be aggregated of other WorkProducts.

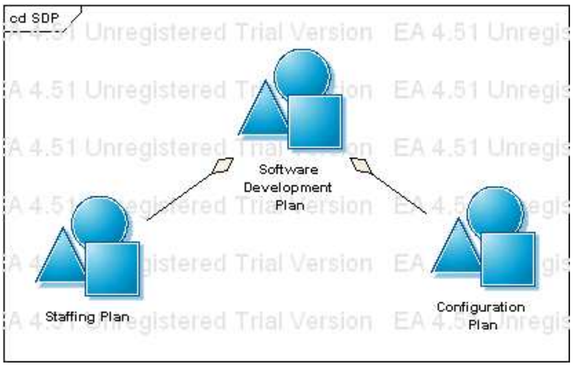


Fig. 1-2 Software Development Plan – WorkProduct that is aggregate of several other documents (Staffing Plan, Configuration Plan, etc.). Diagram painted in EA tool.

WorkProduct can own StateMachine, which describe its states:

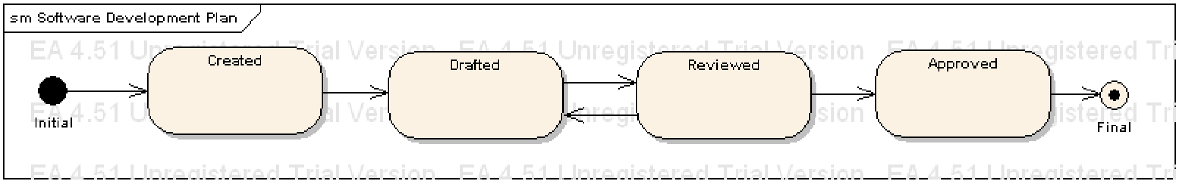


Fig. 1-3 Example of Software Development Plan StateMachine

There can be several kinds of WorkProduct, and some of them have their own icons.

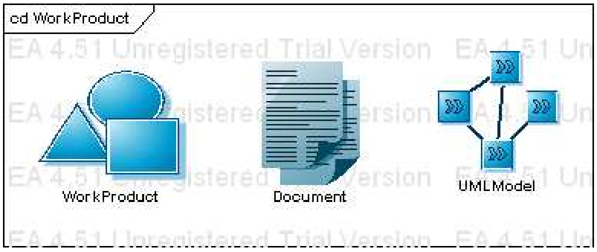


Fig. 1-4 Kinds of WorkProduct

WorkProduct can also have associated role, which is responsible for it.

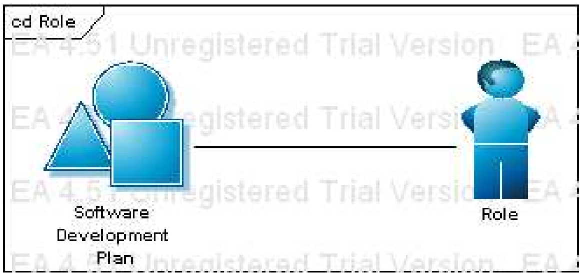


Fig. 1-5 Role responsible for Software Development Plan WorkProduct

WorkDefinition

“WorkDefinition is a kind of Operation that describes the work performed in the process. Its main subclass is Activity, but Phase, Iteration, and Lifecycle (in the Process Lifecycle package) are also subclasses of WorkDefinition. WorkDefinition is not an abstract class, and instances of WorkDefinition itself can be created to represent composite pieces of work that are further decomposed. It has explicit inputs and outputs referred to via ActivityParametr.” [7]

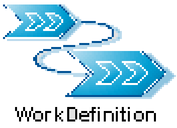


Fig. 1-6 WorkDefinition stereotype symbol (EA tool)

WorkDefinition can be decomposed at UseCase diagram through <<include>> dependency:

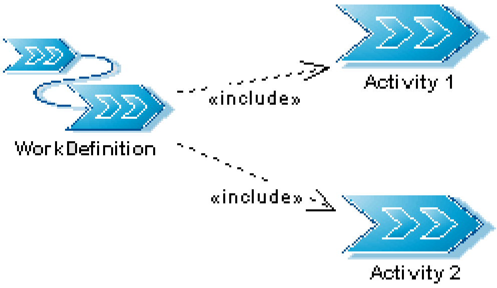


Fig. 1-7 WorkDefinition decomposition

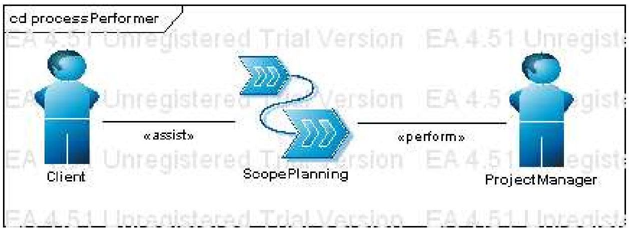


Fig. 1-8 Example of perform and assist stereotypes on associations

Precondition and Goal

“With each WorkDefinition can be associated a Precondition and a Goal. Preconditions and Goals are Constraints, where the constraint is expressed in the form of a Boolean Expression (which is a string) following syntax similar to that of a guard condition in UML. The condition is expressed in terms of the states of the WorkProducts that are the parameters of the WorkDefinition or of an enclosing WorkDefinition.” [SPEM]

Activity and Step

“Activity is the main subclass of WorkDefinition. It describes a piece of work performed by one ProcessRole: the tasks, operations, and actions that are performed by a role or with which the role may assist. An Activity may consist of atomic elements called Steps.”[SPEM]

Fig. 1-9 Activity stereotype symbol (EA tool)

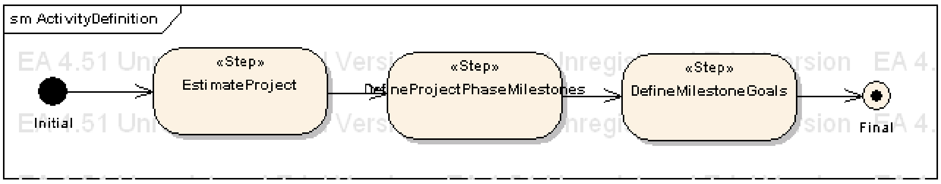


Fig. 1-10 Activity decomposition on steps (Plan Phases And Iterations Activity; source - RUP)

ProcessPerformer and ProcessRole

“A ProcessPerformer defines a performer for a set of WorkDefinitions in a process. ProcessPerformer represents abstractly the whole process or one of its components, and is used to own WorkDefinitions that do not have a more specific owner.” [7]

“ProcessRole is a subclass of ProcessPerformer. ProcessRole defines responsibilities over specific WorkProducts, and defines the roles that perform and assist in specific activities. It was called ‘worker’ in the Rational Unified Process. A ProcessRole is not a person. A given person may be acting in several roles and several persons may act as a single given role.” [7]

Fig. 1-11 Role stereotype symbol (EA tool)

1.5.2.2 Basic Elements

Basic Elements package contains elements that can help in process description.

Guidance

Guidance - Guidance elements maybe associated with ModelElements, to provide more detailed information to practitioners about the associated ModelElement. Most important guidance in aspect of my work are:

- Techniques – detailed, precise algorithm, used to create a work product. Help to define skills required to perform specific types of activities. (Synonyms: procedure, directive)
- Checklist - A checklist is a document representing a list of elements that need to be completed.
- Templates - A Template is a predefined document that provides a standardized format for a particular kind of WorkProduct

1.5.2.3 Process Components

Process Component package contains element, which helps to group elements of processes.

Package

"Package - Just as in UML, a *Package* is a container that can both own and import process definition elements. Between Packages and other ModelElements can exist Categorizes dependency. In such case, package represent category, and elements connected with package via this category are members of this category. Element can belong to several different categories." [SPEM]

Fig. 1-12 ProcessPackage stereotype symbol (EA tool)

ProcessComponent

"A ProcessComponent is a chunk of process description that is internally consistent and may be reused with other ProcessComponents to assemble a complete process. A ProcessComponent imports a non-arbitrary set of process definition elements, modeled in SPEM by ModelElements. Such a set must be self-contained; this means that there are no RefersTo dependencies from within the component to elements not within the component." [SPEM]

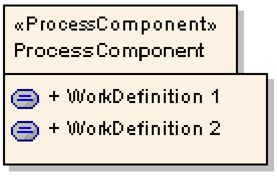


Fig. 1-13 ProcessComponent stereotype (EA tool)

Process
“A Process is a ProcessComponent intended to stand alone as a complete, end-to-end process. It is distinguished from normal process components by the fact that it is not intended to be composed with other components. In a tooling context, the instance of Process is the “root” of the process model, from which a tool can start to compute the transitive closure of an entire process.”[SPEM]

Fig. 1-14 Process stereotype symbol (EA tool)

In other words Process element symbolizes the whole process of software development, and it doesn’t mean the same as conceptual process.

Discipline
“A Discipline is a particular specialization of Package that partitions the Activities within a process according to a common theme”. [SPEM]

1.5.2.4 Process Life Cycle

Process Life Cycle package contains elements, which help describing life cycle of process.

Phase
“A Phase is a specialization of WorkDefinition such that its precondition defines the phase entry criteria and its goal (often called a "milestone") defines the phase exit criteria. Phases are defined with the additional constraint of sequentiality; that is, their enactments are executed with a series of milestone dates spread over time and often assume minimal (or no) overlap of their activities in time.” [SPEM]

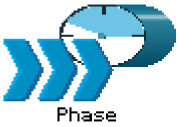


Fig. 1-15 Phase stereotype symbol (EA tool)

Lifecycle
“A process Lifecycle is defined as a sequence of Phases that achieve a specific goal. It defines the behavior of a complete process to be enacted in a given project or program.” [SPEM]

In SPEM as UML profile Lifecycle element is modeled as UseCase element stereotyped as lifecycle, which is composed of UseCases with phase stereotype, through include dependency.

Iteration
"Iteration is a composite WorkDefinition with a minor milestone." [SPEM]

1.5.3 How SPEM elements are presented on UML diagrams

SPEM is metamodel and UML profile. Below table present SPEM stereotypes, and UML elements that respond to SPEM stereotypes.

Table 1-1 SPEM as UML stereotypes

SPEM stereotype	UML element
Guidance	Note
WorkProduct	Class
WorkDefinition	Operation
Activity	Operation
Step	State
ProcessRole	Role
Package	Package
ProcessComponent	Package
Process	Package
Discipline	Package
Phase	Operation
LifeCycle	Operation
Iteration	Operation
Precondition	Constraint
Goal	Constraint

In fact SPEM elements are presented at UML diagrams, and use several UML elements as proxies. Most often used are Activity, Use Case, Package and State chart diagrams.
Bellow tables present, which SPEM elements occur on which UML diagrams, and which UML element serve as proxies for SPEM elements:

Table 1-2 SPEM elements on Activity Diagram

SPEM element	UML proxy element
WorkProduct	ObjectFlowState
WorkDefinition	Activity
Activity	Activity
ProcessRole	Swimlane (not strictly)
Phase	Activity

Table 1-3 SPEM elements on Use Case and Package diagrams

SPEM element	UML proxy element
Guidance	Note
WorkProduct	Class
WorkDefinition	UseCase
Activity	UseCase
ProcessRole	Role
Package	Package
ProcessComponent	Package
Process	Package
Discipline	Package
Phase	UseCase
LifeCycle	UseCase
Iteration	UseCase
Precondition	Constraint
Goal	Constraint

Table 1-4 SPEM elements on State chart diagram

SPEM element	UML proxy element
Step	State
WorkProduct	State

In fact available SPEM profiles don’t obey SPEM specification. For example in tool I have used - Enterprise Architect, all WorkDefinitions on Activity Diagrams were modeled as State stereotypes.

2 Project management activities on the base of PMBOK modeled in SPEM

2.1 PMBOK description

PMBOK abbreviation means “Project Management Body of Knowledge”. It is book that gathers knowledge that considered project management. It consists best project management practices. Book [1] lists PMBOK as one of project management methodologies. In article [8] author states that “the Guide has all the hallmarks of a methodology, and in the absence of anything else is often used as such.” In fact it should not be treated as methodology, because it doesn’t give precise hints how to proceed. PMBOK describes main aspects of project management and is source of widely respected project management knowledge. It describes project management in terms of processes, which occur in project management discipline. It characterizes process as a “series of action bringing about a result”. It describes processes in terms of their inputs, outputs, and tools and techniques, which are applied to inputs to create outputs. Project management processes are characterized as processes that are concerned with describing and organizing the work of the project.

Project management processes and product-oriented processes overlap and interact through the project. (Where product-oriented processes are concerned with specifying and creating the project product).

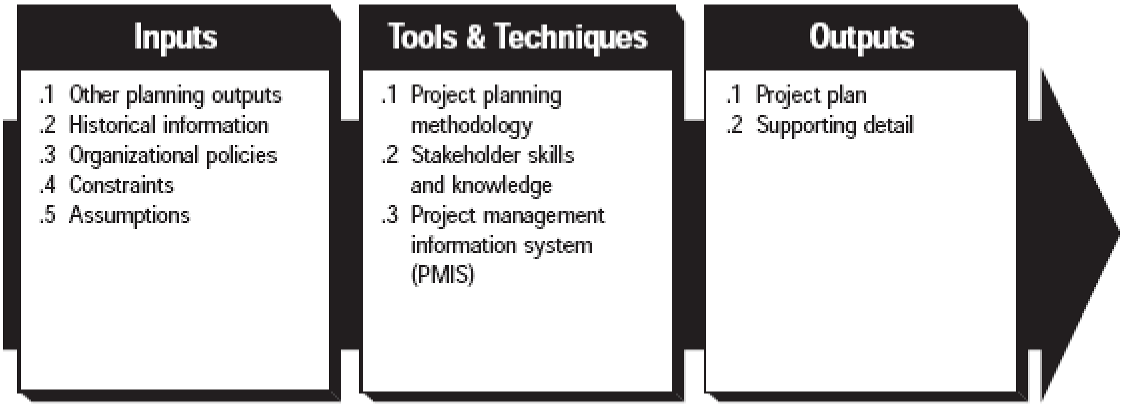

Inputs
.1 Other planning outputs
.2 Historical information
.3 Organizational policies
.4 Constraints
.5 Assumptions
Tools & Techniques
.1 Project planning methodology
.2 Stakeholder skills and knowledge
.3 Project management information system (PMIS)
Outputs
.1 Project plan
.2 Supporting detail

Fig. 2-1 Example of PMBOK Project Plan Development process [source – PMBOK 1.0]

Interaction of processes is presented on below diagram:

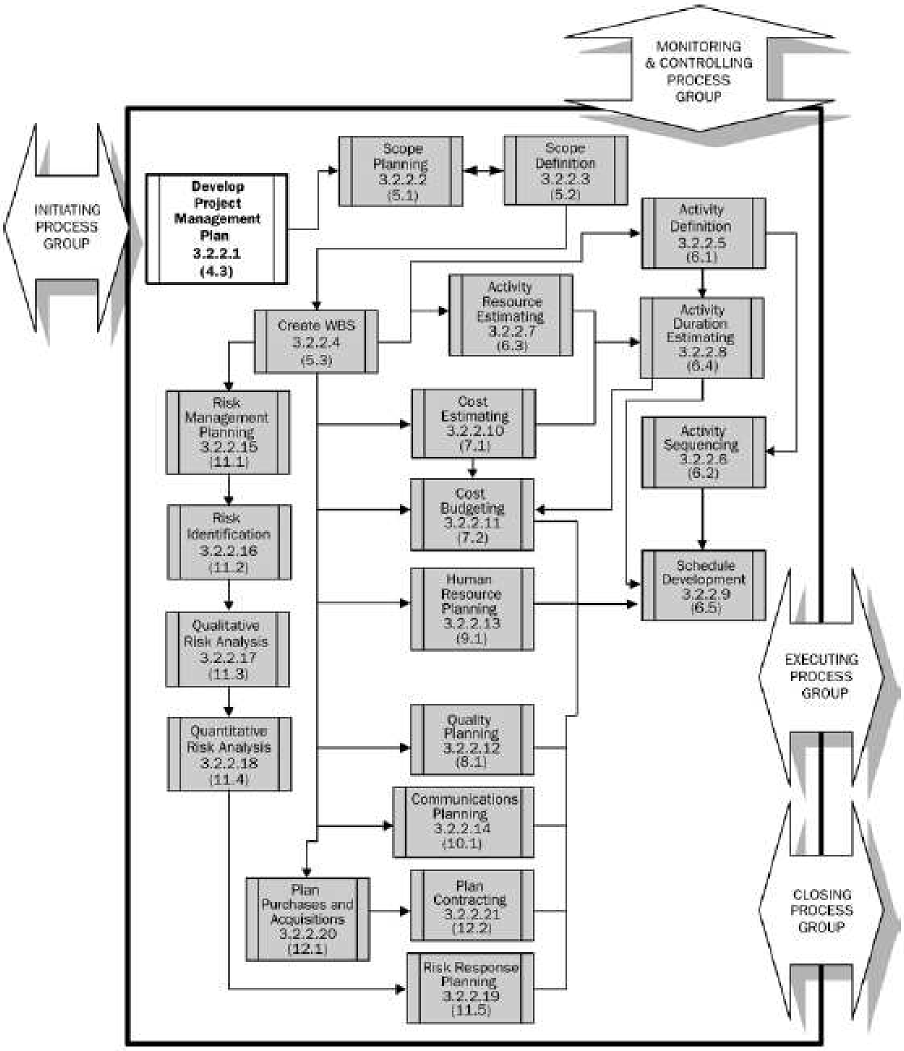

MONITORING & CONTROLLING PROCESS GROUP
INITIATING PROCESS GROUP
Develop Project Management Plan 3.2.2.1 (4.3)
Scope Planning 3.2.2.2 (5.1)
Scope Definition 3.2.2.3 (5.2)
Activity Definition 3.2.2.5 (6.1)
Create WBS 3.2.2.4 (5.3)
Activity Resource Estimating 3.2.2.7 (6.3)
Activity Duration Estimating 3.2.2.8 (6.4)
Risk Management Planning 3.2.2.15 (11.1)
Cost Estimating 3.2.2.10 (7.1)
Activity Sequencing 3.2.2.6 (6.2)
Risk Identification 3.2.2.16 (11.2)
Cost Budgeting 3.2.2.11 (7.2)
Schedule Development 3.2.2.9 (6.5)
Human Resource Planning 3.2.2.13 (9.1)
Qualitative Risk Analysis 3.2.2.17 (11.3)
EXECUTING PROCESS GROUP
Quantitative Risk Analysis 3.2.2.18 (11.4)
Quality Planning 3.2.2.12 (8.1)
Communications Planning 3.2.2.14 (10.1)
Plan Purchases and Acquisitions 3.2.2.20 (12.1)
Plan Contracting 3.2.2.21 (12.2)
CLOSING PROCESS GROUP
Risk Response Planning 3.2.2.19 (11.5)

Fig. 2-2 PMBOK diagram that presents process interactions [source PMBOK 3.0]

There are two ways of organizing processes, which PMBOK presents.
First way is organizing processes into process groups, and there is five groups of processes, according to their occurrence and role in the project:

1. Initiating
2. Planning
3. Executing
4. Controlling
5. Closing

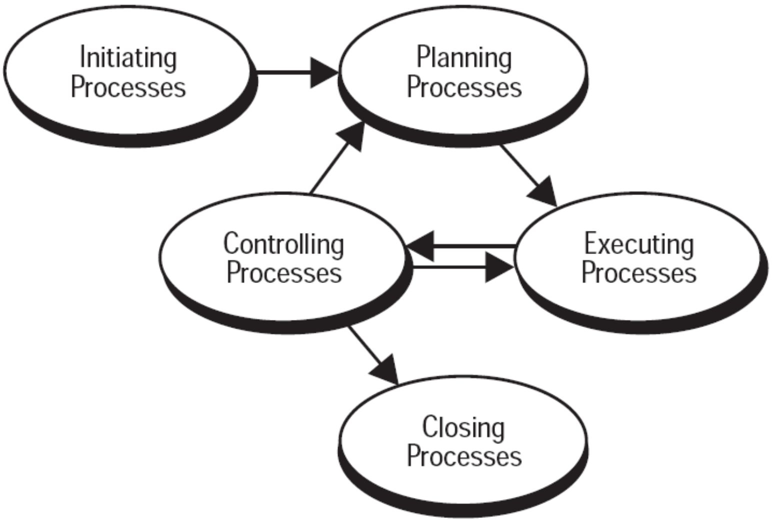


Fig. 2-3 Presents grouping processes into process groups according to their occurrence and role in the project [source – PMBOK 1.0]

Second way is organizing processes according to common theme, and in PMBOK this process grouping is called knowledge area grouping. There are nine knowledge areas:

1. Project Integration Management
2. Project Scope Management
3. Project Time Management
4. Project Cost Management
5. Project Quality Management
6. Project Human Resource Management
7. Project Communications Management
8. Project Risk Management
9. Project Procurement Management

The processes described within knowledge areas interact with each other. They are linked together by results they produce. Among the central process groups (Planning, Controlling, Executing) the links are iterated. Planning provides plans for executing processes, and then provides updates to the plan, as project progress. Project management processes are not discrete, one-time events. They are overlapping and cyclic activities which occur at varying levels of intensity through each phase of project.

2.2 PMBOK processes and knowledge areas

In this chapter I would like to describe PMBOK processes and several PMBOK knowledge areas – the most important in context of my work.

Simple mapping between PMBOK processes and activities is not always possible. Whereas for initiating, planning and closing process groups PMBOK processes can simply map to activities, for controlling and executing processes it is not always possible. PMBOK is described in terms of knowledge areas, and it is not methodology. It doesn't provide description of activities that should occur during project management, only provides description of processes, which concern certain knowledge areas. Bellow is provided short description of knowledge areas, and processes within them, that I've analyzed

As described earlier, PMBOK knowledge areas organize project management processes according to common theme, to present one aspect of project management.

2.2.1 Project Scope Management

To Project Scope Management belong processes:

- Scope Planning
- Scope Definition
- Create WBS
- Scope Verification
- Scope Control

Scope Planning, Scope Definition and Create WBS processes are planning activities. They can be simply modeled as SPEM activities. They generally are responsible for creating project scope statement. In several methodologies this processes are mapped to requirements analysis activities, because they define scope of system.

Scope Verification process verifies weather produced scope of system responds to required scope. PMBOK describe Inspections, Reviews, and Walkthroughs as techniques of Scope Verification process. So Scope Verification could be also modeled on Activity Diagram as Inspection Activity (Review, Walkthrough etc.). Occurrence of this process is connected with Life Cycle – Scope Verification process generally occur when deliverable is ready, so at the end of phase or iteration. But it also can occur in case of interim deliverables. Example of Scope Verification activities on model is presented on fig. x *** Planned Scope Verification activities are also part of Quality Assurance process.

Scope Control process "is concerned with influencing the factors that create project scope changes and controlling the impact of this changes". It also concern managing actual changes. This process is integrated with other control processes. Its main techniques are Change control system and Configuration management system. These systems are described as "procedures by which the project scope and product scope can be changed". This process should be included into Integrated Change Control Process. (Example of Change Control process from RUP is presented at fig. x ***)

2.2.2 Project Time Management

To Project Time Management belong processes:

- Activity Definition
- Activity Sequencing

- Activity Resource Estimating
- Activity Duration Estimating
- Schedule Development
- Schedule Control

Activity Definition, Activity Sequencing, Activity Resource Estimating, Activity Duration Estimating and Schedule Development are processes of Project Time Management that generally take place during project planning. They can occur also in occasion of Change Requests, which is after-effect of change, preventive or corrective action. Schedule Control generally is responsible for:

- "Determining current status of schedule
- Influencing the factors that create schedule changes
- Determining that project schedule has changed
- Managing actual changes as they occur" [PMBOK 3.0]

Several controlling and analysis tools and techniques are used to achieve these objectives, for example variance analysis.

2.2.3 Project Cost Management

Project Const management consists of processes:

- Cost Estimating
- Cost Budgeting
- Cost Control

Similarly as for previous knowledge areas, also here cost is estimated and budgeted (Cost Estimating and Cost Budgeting) on the base of project scope and schedule, and then is controlled and updated during execution phase. Cost Control is similarly as in previous described knowledge area process responsible for controlling and monitoring project costs, determining if changes are needed and managing changes.

2.2.4 Project Quality Management

Project Quality Management is knowledge area that concerns both – management and development processes. It consists of:

- Quality Planning,
- Quality Assurance and
- Quality Control processes.

Quality Planning process is responsible for planning quality into project in terms of activities processes or best practices that ensures required quality.

Quality Assurance is general term for performing activities that are planned in project management plan.

Quality Control concerns controlling quality aspects of project, and it use several techniques to compare required quality metrics with achieved results.

Bellow model present model pattern, where in case of negative quality control Repair is needed to deliverable.

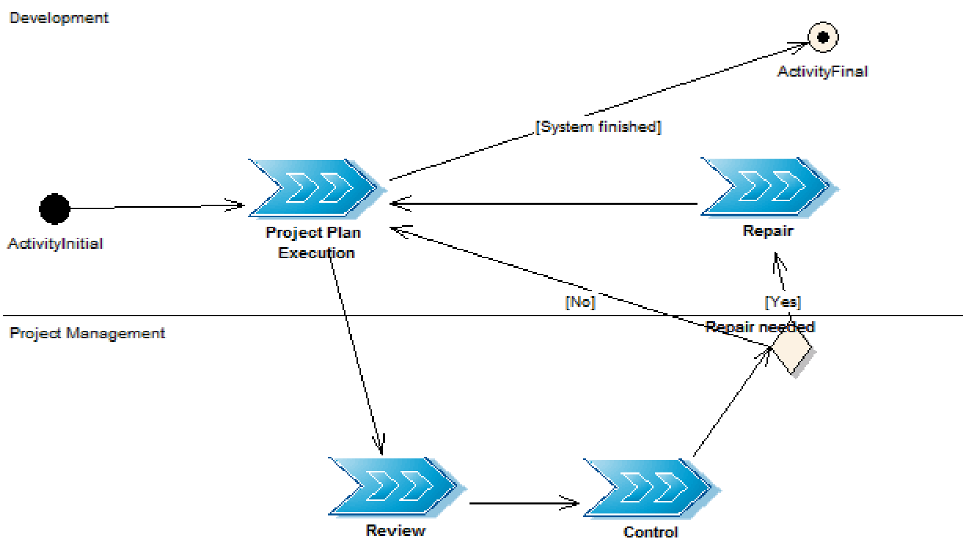


Fig. 2-4 Model pattern for Repair

2.2.4.1 How Project Quality Management knowledge area relates to plan-do-check-act cycle:

Processes described in Project Quality Management knowledge area actually are in harmony with plan-do-check-act cycle (PDCA cycle). Quality is fist planned in Quality Planning process. Next, after planning process occurs Direct and Manage Project Execution (Project Plan Execution in PMBOK 1.0) process in which project product is prepared and which responds to “do” part of PDCA cycle. During Project Plan Execution occurs Perform Quality Assurance process, which is execution of planned activities, and which together with Perform Quality Control process respond to “check” part of PDCA cycle. Next it comes to “act” part of the cycle, which in PMBOK can be: Corrective action, Preventive action, Repair, Plan updates and Organization process assets updates (“Process adjustment” in PMBOK 1.0).

2.2.4.2 How Project Quality Management knowledge area relates to CMM:

From PMBOK point of view, increased process maturity allow for higher control over quality. So during Quality Planning and other planning PMBOK processes, project manager can take into consideration CMM processes, and make plan that would take them into consideration.

From CMM point of view, it requires certain quality management processes to be present at certain maturity level – it requires Software Quality Assurance at 2 level, and Software Quality Management at 4 level.

Software Quality Assurance (according to CMM) should assure quality in project through planning – quality should be planned in, and appropriate processes should be chosen to perform within life cycle, for example quality audits and reviews should be planned and next, as part of Quality Assurance Process they should be performed. This responds to PMBOK Quality Planning and Perform Quality Assurance processes.

Software Quality Management (according to CMM) “The purpose of Software Quality Management is to develop a quantitative understanding of the quality of the project's software products and achieve specific quality goals. Software Quality Management involves defining quality goals for the software products, establishing plans to achieve these goals, and monitoring and adjusting the software plans, software work products, activities, and quality goals to satisfy the needs and desires of the customer and end user for high quality products." So Software Quality Management according to CMM consider constant ongoing measurements of quality metrics, and in case of bad result taking corrective actions, or adjusting processes to fit quality requirements. It also requires quantitative planning for quality requirements. This CMM process responds to PMBOK Quality Planning, in which Quality metrics and Quality baseline are established, Perform Quality Assurance and Perform Quality Control processes, in which processes and their outputs are measured.

2.2.4.3 Quality Project Management summary

It can be stated, that Quality Assurance process actually includes other processes and activities, and performance of this activities assures quality. It includes both – management and development activities, and model of Quality Assurance should contain Quality Planning activity, all kind of Review and Test activity. Additionally it should contain Quality Control activity, which should be included into Monitor and Control WorkDefinition, and which should control quality metrics.

Above comparison of CMM and PMBOK in area of quality can be base for further work. For example

- From PMBOK point of view, to assure high quality process engineer could create CMM process models in SPEM. Project manager in Quality Planning process could choose this models, which would assure highest quality.
- From CMM point of view there could be created project management process models, that would respond to certain CMM levels, and company that would like to reach certain level would need to have certain management processes on place
- Finally project management maturity process model could emerge and could be modeled in SPEM. Actually some management maturity process models are available, for example described in [15]

2.2.5 Project Communication Management

This knowledge area contains processes responding to planning and spreading project related information. From my work point of view Performance Reporting process is important. Important inputs to this process are Work performance information from Direct and Manage Project Execution. Important outputs from this process are Performance reports, which serve as input to several previously described controlling processes, corrective actions and change requests. It generally should provide information on scope, schedule, cost and quality. Many project requires also information on risk and procurement.

2.2.6 Project Risk Management

Project Risk Management consists of several processes, which generally can be rated among planning and monitoring processes:

- Risk Management Planning
- Risk Identification
- Qualitative Risk Analysis
- Quantitative Risk Analysis
- Risk Response Planning
- Risk Monitoring and Control

Risk planning according to PMBOK constitutes Risk Management Planning, Risk Identification, Qualitative Risk Analysis, Quantitative Risk Analysis and Risk Response

Planning. This activities occur also within project life cycle in case of identifying new risk, or in case of change requests, where risk connected with each change need to be analyzed.

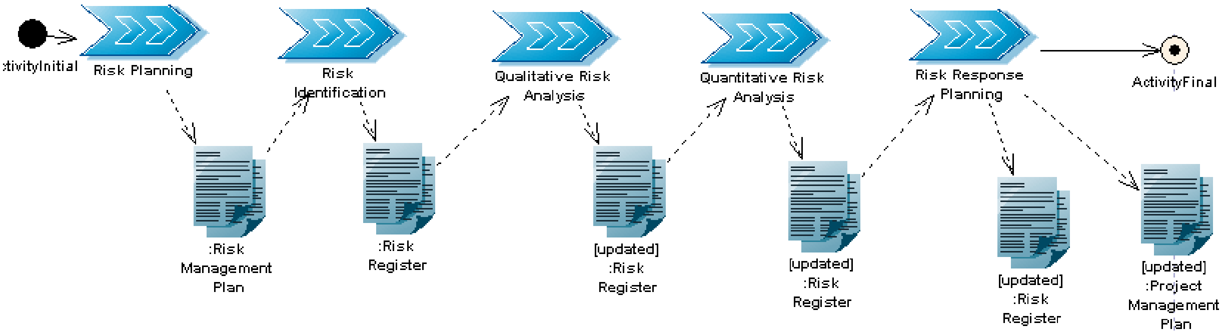


Fig. 2-5 Risk planning process model

Risk Management Planning is the basic process of risk management. Steps of risk planning process are presented at bellow model:

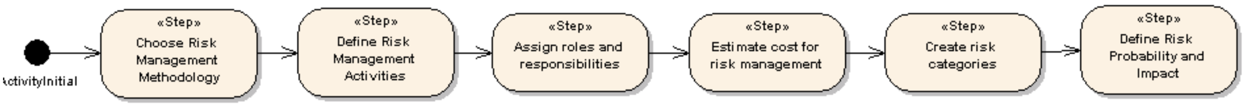

ctivityInitial
«Step» Choose Risk Management Methodology
«Step» Define Risk Management Activities
«Step» Assign roles and responsibilities
«Step» Estimate cost for risk management
«Step» Create risk categories
«Step» Define Risk Probability and Impact

Fig. 2-6 Example Risk Planning process

At the beginning risk methodology should be chosen to treat risk. This could mean choosing one of methodologies, for example XP or Spiral model, which approach to risk is described further. So this stage has big influence on how model for development will look like. Bellow (Risk management and methodology point) this issue is described more precisely.
Further risk activities are identified and included into overall project plan, roles are assigned to this activities and cost are estimated. Further risk categories are identified, and probability of risk occurrence and its impact is defined. Output from this process is Risk Management Plan. Next process described in PMBOK is Risk Identification. This process is an iterative process, because new risk may become known later in project life cycle, so it should be taken into consideration during creating life cycle process model.

Further according to PMBOK risk is monitored and controlled (Risk Monitoring and Controlling process). Project Manager has several tools and techniques to perform risk monitoring and control. For example he can place at project plan Risk Audit activities (which is one of the technique). Generally project manager should monitor if sign of risk occur, and Risk Monitoring and Control activity should be placed in Monitor and Control WorkDefinition. In case of signs of risk occurrence project manager should undertaken preventive action.

Influence of risk management on process modeling
Risk planning at the beginning of the process also involves choosing appropriate methodology and appropriate processes to treat risk. This can have influence on which life cycle for project will be chosen or which processes occur in project. This has influence on process models that will be used in project.

2.2.7 Project Integration Management

Project integration management knowledge area is most important part of project management. The objective of integration management is to integrate all aspects of project management. For example quality aspects influence time management, because assuring quality requires placing review activities on plan, what extend project duration. This could have influence on risk, for example possibility of missing delivery data would increase. PMBOK 3.0 lists seven processes within Project Integration Management:

1. Develop Project Charter
2. Develop Preliminary Project Scope Statement
3. Develop Project Management Plan
4. Direct and Manage Project Execution
5. Monitor and Control Project Work
6. Integrated Change Control
7. Close Project

Develop Project Charter and Develop Preliminary Project Scope Statement are concerned with starting project. At the beginning some data are needed to create plans, which could be baseline for project at this stage, until more detailed data will be available. This is the role of this processes.

Develop Project Management Plan is process, which integrate all planning processes. Planning process can be very complicated task. It is impossible to present all process dependencies on one process model.

Monitor and Control Project Work is process, which concerns collecting, measuring and disseminating performance information.

Integrated Change Control is the process of integration of change processes. Change integration is, similarly as planning, very complex process. It generally can concern changes to scope, budget, schedule, quality requirements procedures or policies. Change can arise from several sources – from execution of project plan (for example scope change request), from performance reports (for example schedule change request), from controlling processes, and from other changes (for example change in scope could require change in schedule). To capture this complexity, PMBOK created Integrated Change Control process. Integrated Change Control concerns also Corrective Actions, Preventive Actions and Repairs. This actions are in fact changes to some aspect of project – for example to prevent risk arising schedule can be extended, or schedule must be changed in case of repair. Changes, Corrective Actions and Preventive actions requires to some planning processes to be repeated, to create updated Project Management Plan.

Direct and Manage Project Execution is in PMBOK process responsible for developing project deliverable. It can be treated as proxy for the whole development process. In PMBOK it delivers to other management processes deliverables and performance information that serves as inputs to management control processes.

2.3 PMBOK processes modeled as SPEM ProcessComponents

On the base of processes described in PMBOK, project management process models can be created. There are several ways and possibilities of how project management process models can be presented, for example:

1. They can be presented as project management methodology, and can be modeled as such.
2. They can be presented as framework, and certain processes can be modeled as ProcessComponents.

This both possibilities don't exclude each other. For example from methodology ProcessComponent can be distinguished, and from ProcessComponents methodology can be created.

Bellow I propose how to model project management processes as ProcessComponents. ProcessComponents were described in chapter that relates SPEM metamodel. There is constrains that elements within process component can't refer to work products from other process component. Process component must be internally consistent.
At this level it depend on process engineer how he will model process components. PMBOK grouped naturally processes into process groups, which are good candidates, to be process components. This approach is presented further.

Presented ProcessComponents have interfaces in form of WorkProducts, which need to be delivered from development processes to management processes. Composition of several ProcessComponents is called *unification* by SPEM specification. SPEM states, that in process of unification in case ProcessComponents came from different families, sometimes the unification would require human intervention, that can consists of rewriting of the elements and possibly associating elements to be unified. But I won't be considering such cases.

Bellow I presented project management components, and description how they relate to PMBOK.

Management components:

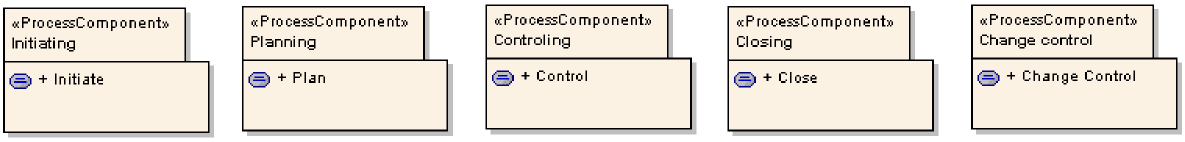


Fig. 2-7 Project management ProcessComponents

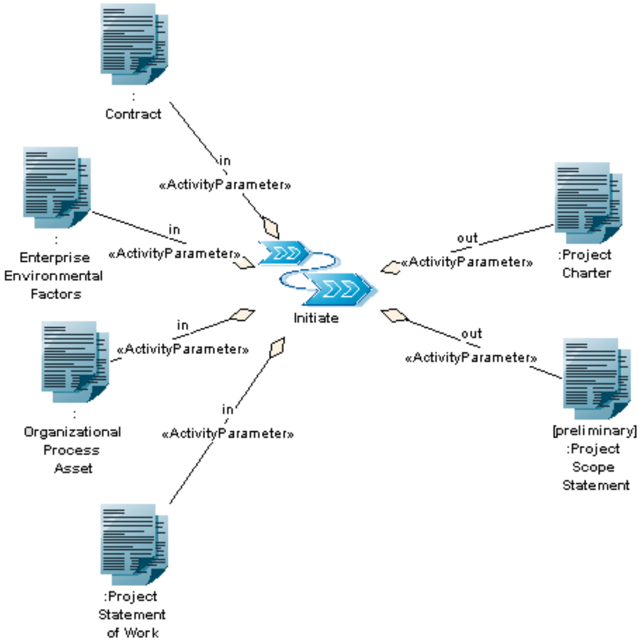


Fig. 2-8 Inside of Initiating ProcessComponent

Initiating ProcessComponent consists of Initiate WorkDefinition, which take Contract, Enterprise Environmental Factors, Organizational Process Asset and Project Statement of Work as inputs, and deliver Project Charter and Preliminary Project Scope Statement as output.

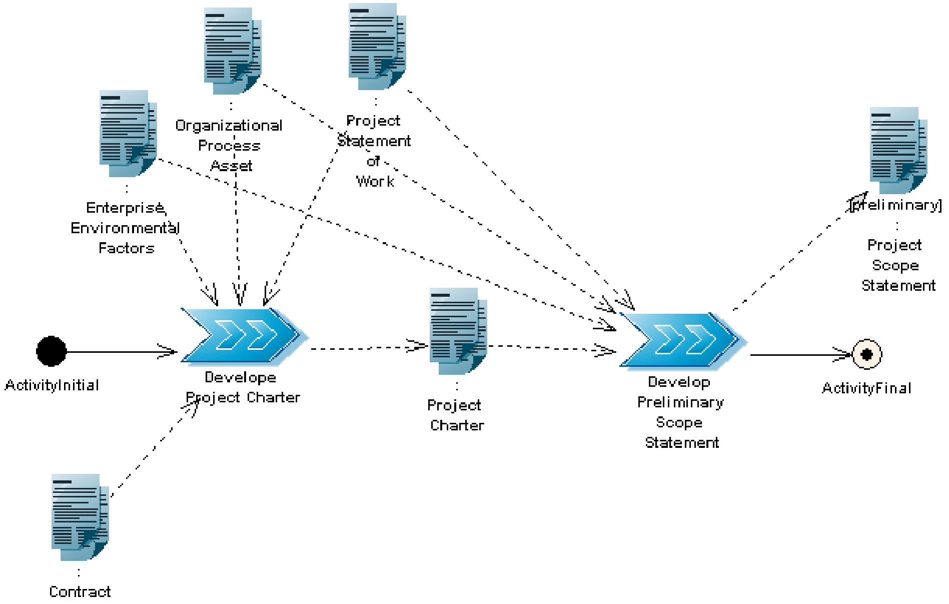


Fig. 2-9 Activity Diagram related to Initiation WorkDefinition.

Initiation WorkDefinition contains Develop Project Charter and Develop Preliminary Scope statement, which are performed at the beginning of the project. In case that project starts subsequent phase Verify Project Charter and Scope Statement activity is performed.

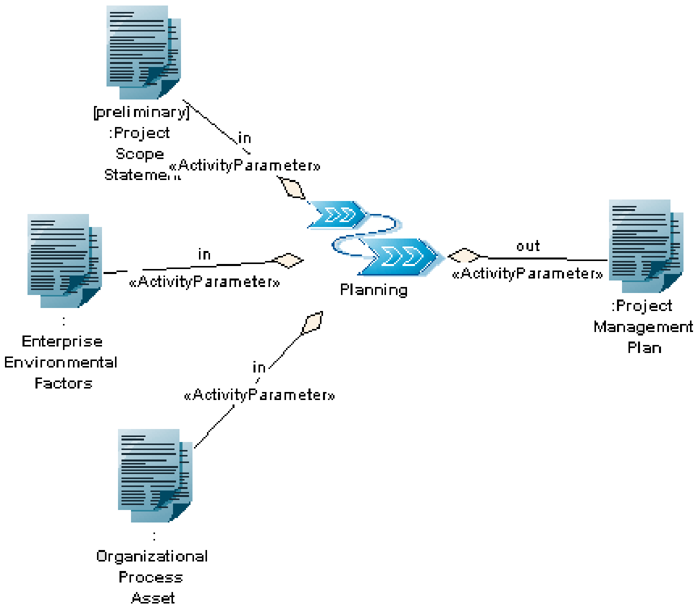


Fig. 2-10 Inside of Planning ProcessComponent

Planning ProcessComponent consists of Planning WorkDefinition, which takes Preliminary Project Scope Statement, Enterprise Environmental Factors and Organizational Process Asset as inputs, and delivers Project Management Plan as output. One of possibility to present planning WorkDefinitions is to present various project planning activities as parallel activities, because interactions between this activities during project plan creation can be too complicated.

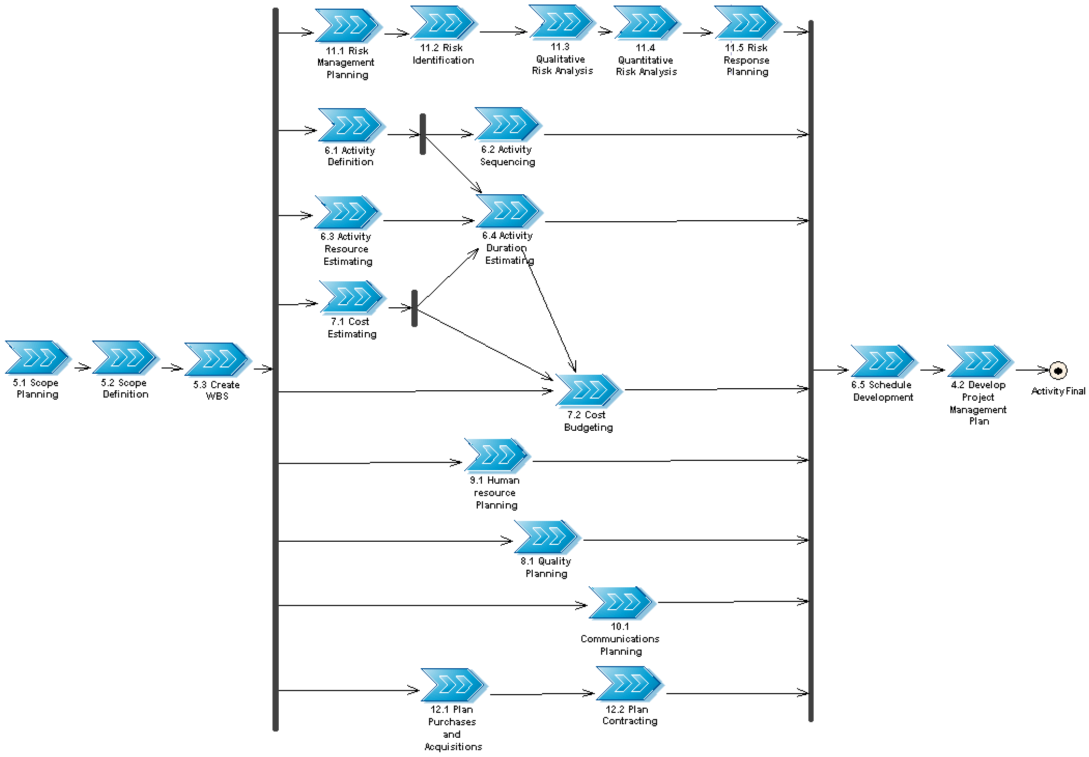

11.1 Risk Management Planning
11.2 Risk Identification
11.3 Qualitative Risk Analysis
11.4 Quantitative Risk Analysis
11.5 Risk Response Planning
6.1 Activity Definition
6.2 Activity Sequencing
6.3 Activity Resource Estimating
6.4 Activity Duration Estimating
7.1 Cost Estimating
5.1 Scope Planning
5.2 Scope Definition
5.3 Create WBS
7.2 Cost Budgeting
6.5 Schedule Development
4.2 Develop Project Management Plan
ActivityFinal
9.1 Human resource Planning
8.1 Quality Planning
10.1 Communications Planning
12.1 Plan Purchases and Acquisitions
12.2 Plan Contracting

Fig. 2-11 Activity Diagram that relate to Planning WorkDefinition. (WorkProducts that are inputs and outputs are not presented)

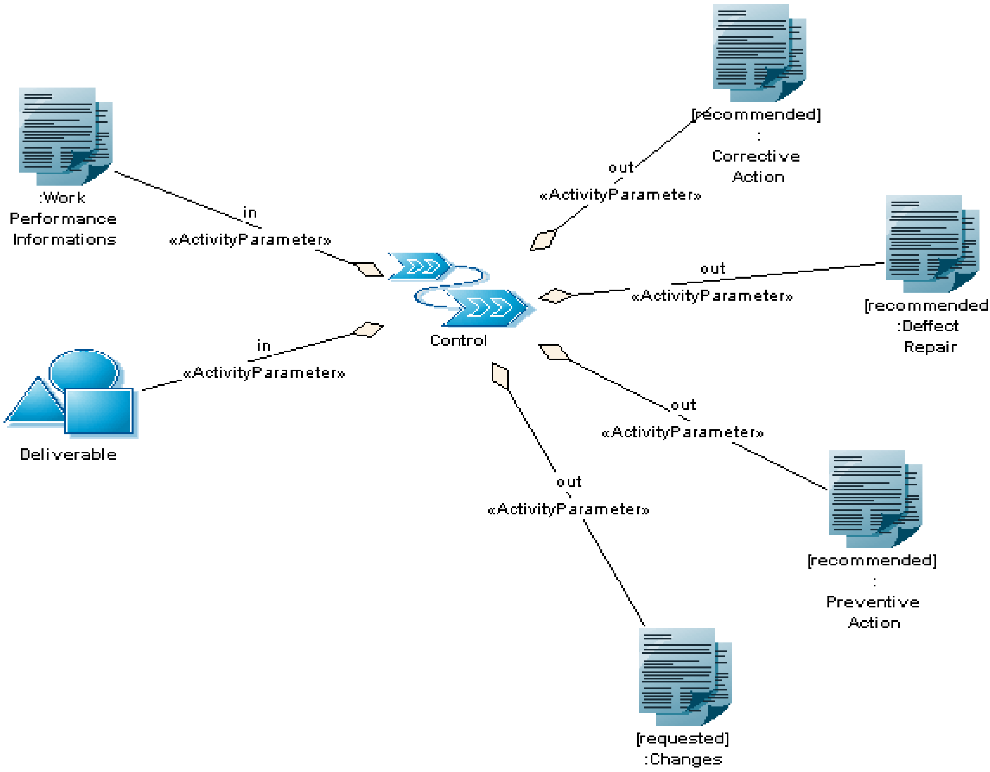


Fig. 2-12 Inside of Controlling ProcessComponent

Controlling ProcessComponent contains Monitoring and Control WorkDefinition, which takes Work Performance Information and Deliverables as input, and delivers recommended Defect Repair, recommended Corrective Action, recommended Preventive Action and requested Changes as output.

Monitoring and Control WorkDefinition contains Review activity, which is mapped to Performance Report PMBOK process, and also to Inspection and Defect Repair Review that are techniques of Quality Control, and Status Meetings which is technique of Risk Monitoring and Control. Output from Review activity – Performance Report is input to several control activities that are performed by project manager. Within Monitoring and Control WorkDefinition I have place Schedule, Cost and Scope Reporting activities, which belong also to Communication Management process, but their role is to track schedule, cost and scope, and require action in case of negative results. This is similar task as for Risk Monitoring and Control and Quality Control Activities. Outputs from this control activity are presented at below model, and they are used as inputs to Integrated Change Control WorkDefinition.

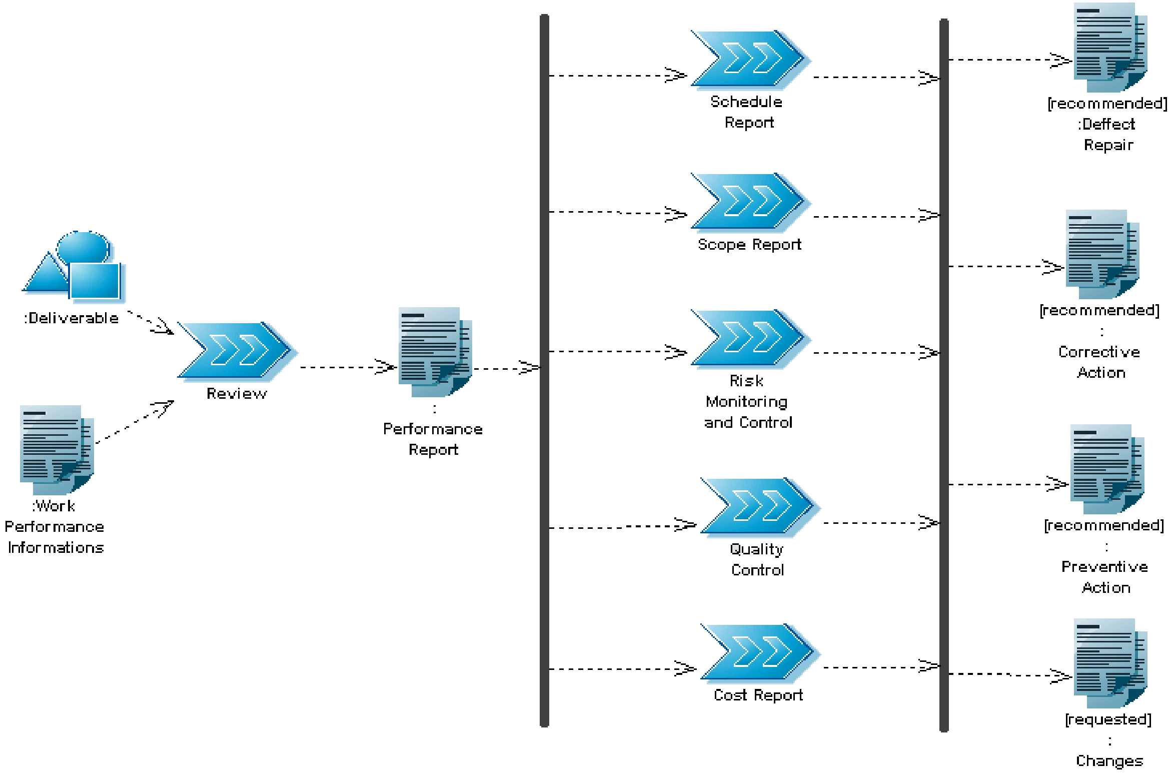

:Deliverable
:Work Performance Informations
Review
: Performance Report
Schedule Report
Scope Report
Risk Monitoring and Control
Quality Control
Cost Report
[recommended] :Deffect Repair
[recommended] : Corrective Action
[recommended] : Preventive Action
[requested] : Changes

Fig. 2-13 Activity Diagram that relate to Monitoring and Controlling WorkDefinition

Change Control ProcessComponent contains Change Control WorkDefinition, which can take recommended corrective action, preventive action, change request or defect repair as input, and approve or reject them.

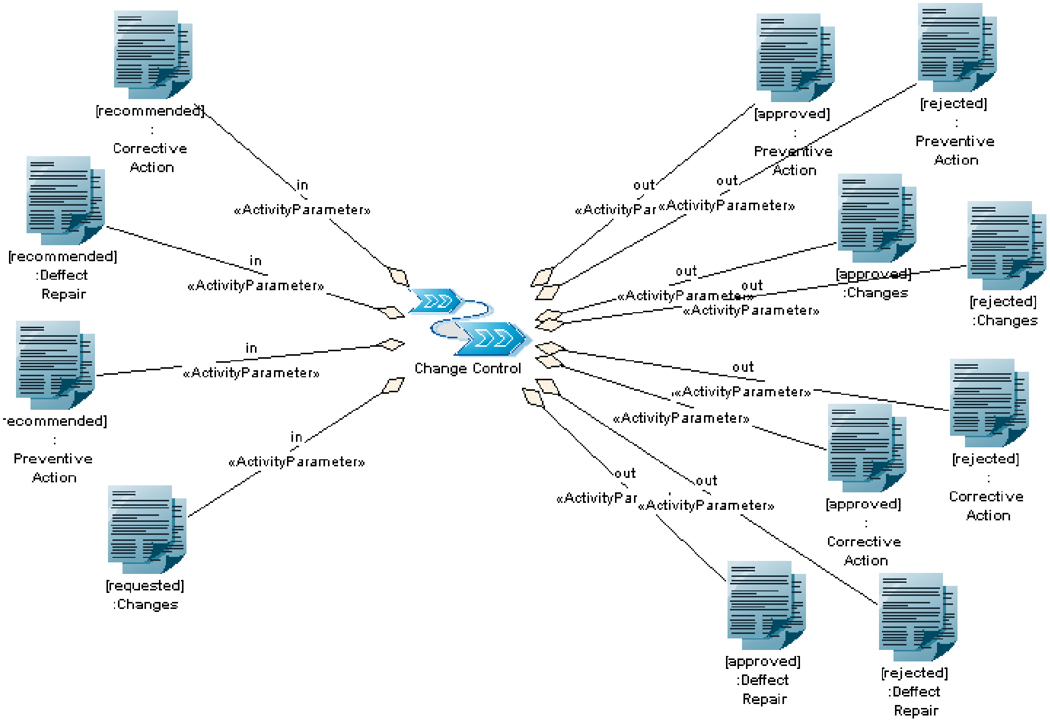

[recommended] : Corrective Action
[recommended] :Deffect Repair
recommended] : Preventive Action
[requested] :Changes
in
«ActivityParameter»
Change Control
out
[approved] : Preventive Action
[rejected] : Preventive Action
[approved] :Changes
[rejected] :Changes
[approved] : Corrective Action
[rejected] : Corrective Action
[approved] :Deffect Repair
[rejected] :Deffect Repair

Fig. 2-14

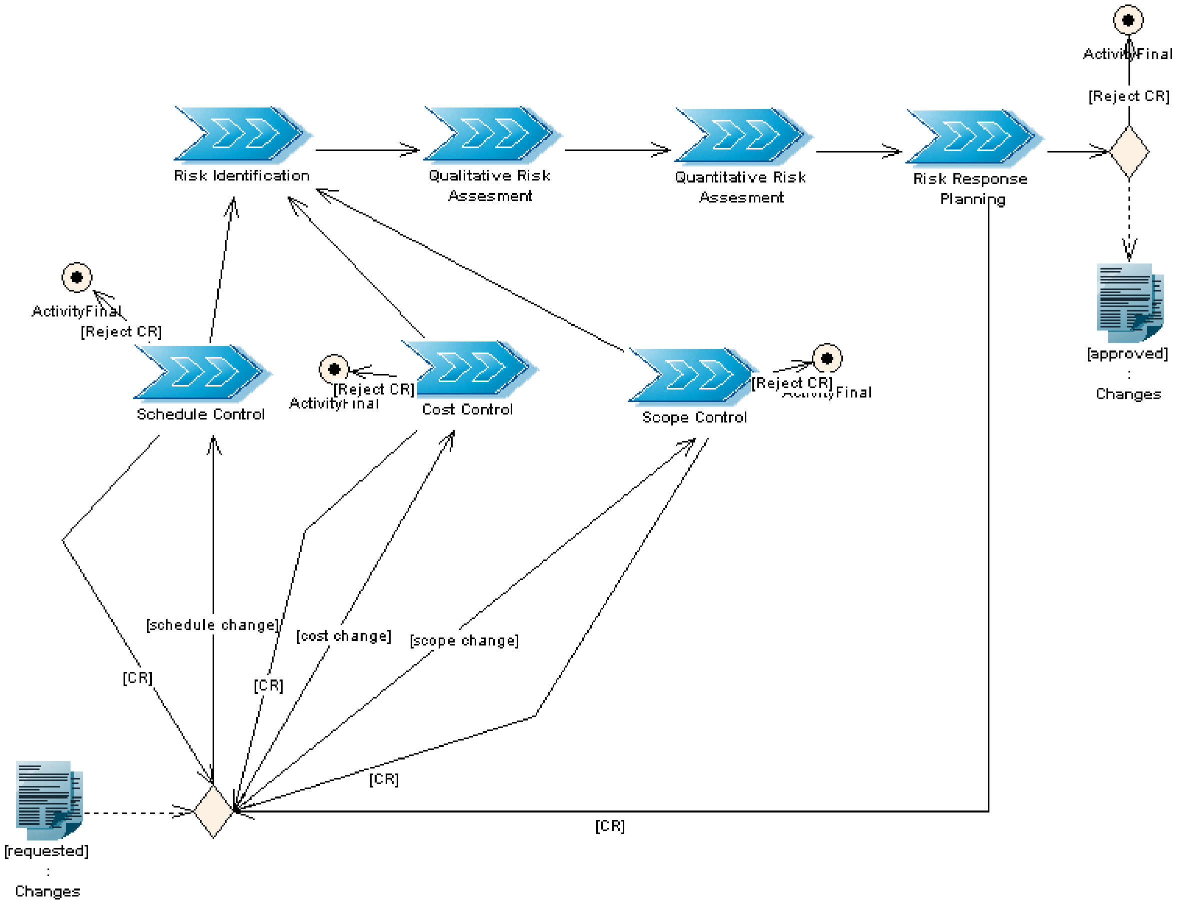


Fig. 2-15 Activity Diagram that relate to Integrated Change Control WorkDefinition

Example Change Control activity diagram is presented at above model. This diagram present Schedule, Cost and Scope Control activities, which in PMBOK are responsible for controlling impact of change on project, and approving changes as required. On the diagram are not presented Preventive and Corrective actions and Defect Repairs, which are also inputs and outputs to Integrated Change Control. In case of requested change, it is determine which aspect of project it concerns, and conducting analysis of this aspect. Such analysis requires taking into consideration several alternatives, and can result in change request to other aspect of project. Finally risk for every proposed change is identified, and change is approved or rejected.

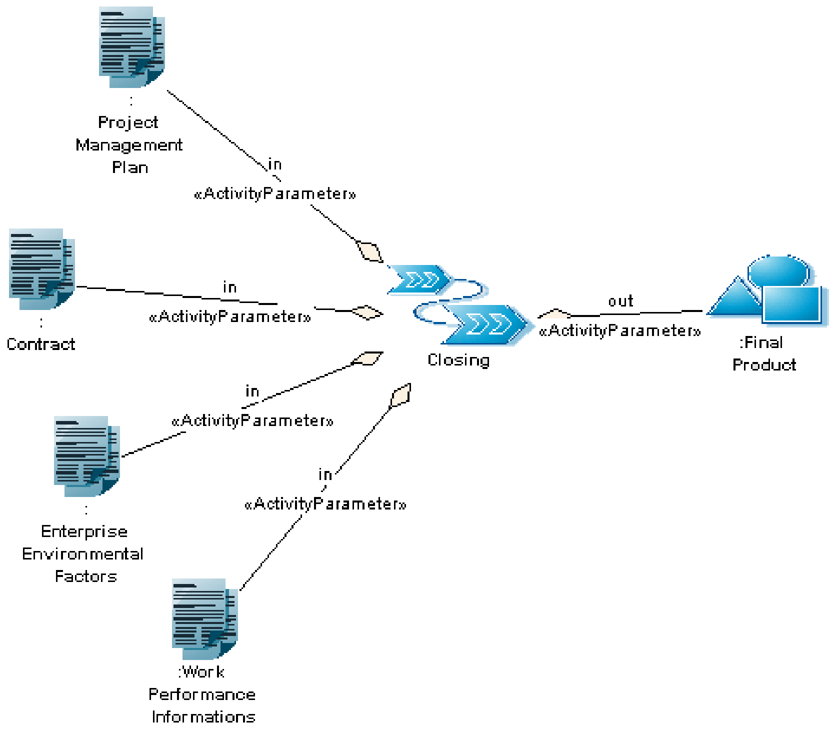


Fig. 2-16 Inside of Closing ProcessComponent

Closing ProcessComponent contains Closing WorkDefinition, which takes Project Management Plan, Contract, Enterprise Environmental Factors as inputs, and in case this process in closing process for iteration, and not project, it takes Work Performance Information as input. It delivers Final Product as output.

Below is presented activity diagram for Closing WorkDefinition. In case of project closing Contract Close activity is performed. In case of closing phase, Scope Verification is performed.

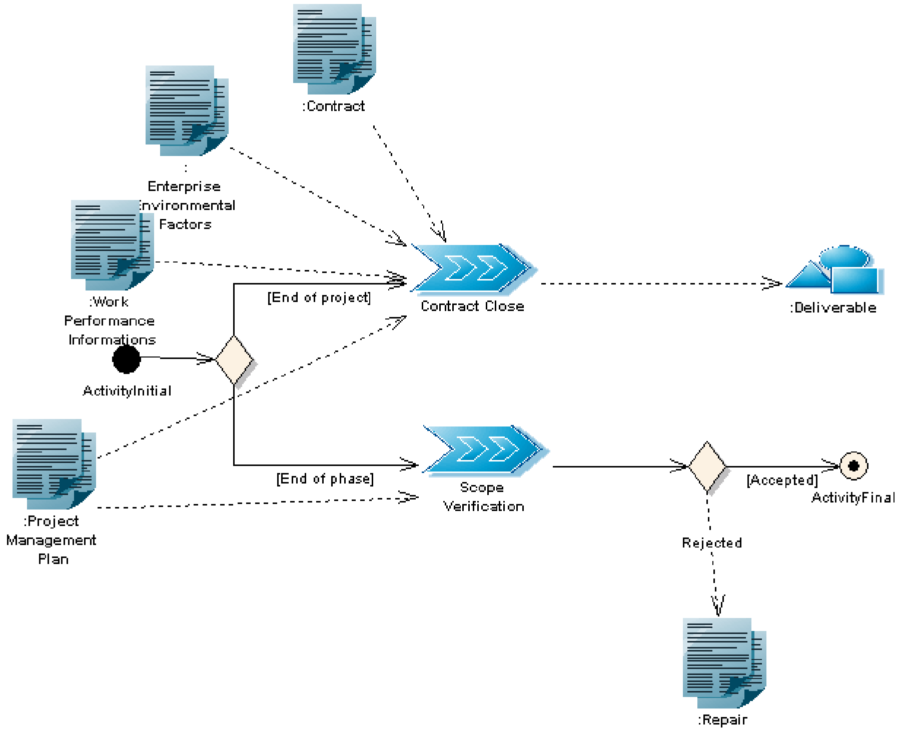


Fig. 2-17 Activity Diagram of Closing WorkDefinition.

Above modeled project management components can now be used in projects. It doesn't mean that all processes that were defined need to by use in projects. In fact, project manager and project team need to customize processes, and choose this, that is most suitable for certain project.

2.4 Problems with modeling project management activities in SPEM

There arise problem concerned modeling project management processes. This is because controlling and executing project management activities involves interactions with development processes. PMBOK describe several techniques that enable such interactions. (PMBOK processes are described in next chapter) most of them described in Performance Reporting process (within Project Communication Management knowledge area), but also in Scope Verification, Quality Control and Quality Assurance processes.

Performance reporting process defines Status review meeting, and several systems, that enable project related information (mainly on scope, schedule, time and quality) to project manager and interested stakeholders.

Scope Verification and Quality Control as a source of information defines Inspection activity. This activity doesn't have to be performed by project manager. In fact it is probably not project manager who will perform this activity, but project manager can earn information about scope and quality on base of inspection information.

Quality Assurance defines Quality Audits activity that serves gathering information about quality. It is “structured, independent review to determine weather project activities comply with organizational and project policies, processes and procedures. The objective of a quality audit is to identify inefficient and ineffective policies, processes and procedures in use on the project. [...] Quality Audits can be scheduled or at random [...] Quality audits confirm the implementation of approved change requests, corrective actions, defect repairs, and preventive actions” [PMBOK 3.0]

There are several problems connected with modeling described above processes, and there are several proposed solutions.

First problem is how to present Scope Verification activities. Scope Verification activities belong to kind of activities that takes place, when certain deliverable is ready. This issue is usually presented at software development life cycle model as goal of phase. Scope Verification activities can also emerge inside phase, as intermediate deliverables are ready. Bellow is presented example of Software Architectural Design Phase Activity Diagram. This model is based on ISO/IEC 12207 standard. This standard among the others describe Conduct Joint Review and Evaluation activities as Scope Verification and Quality Assurance activities. These activities are presented on diagram, and it is possible to present them on such diagram because they have exactly defined occurrence time.

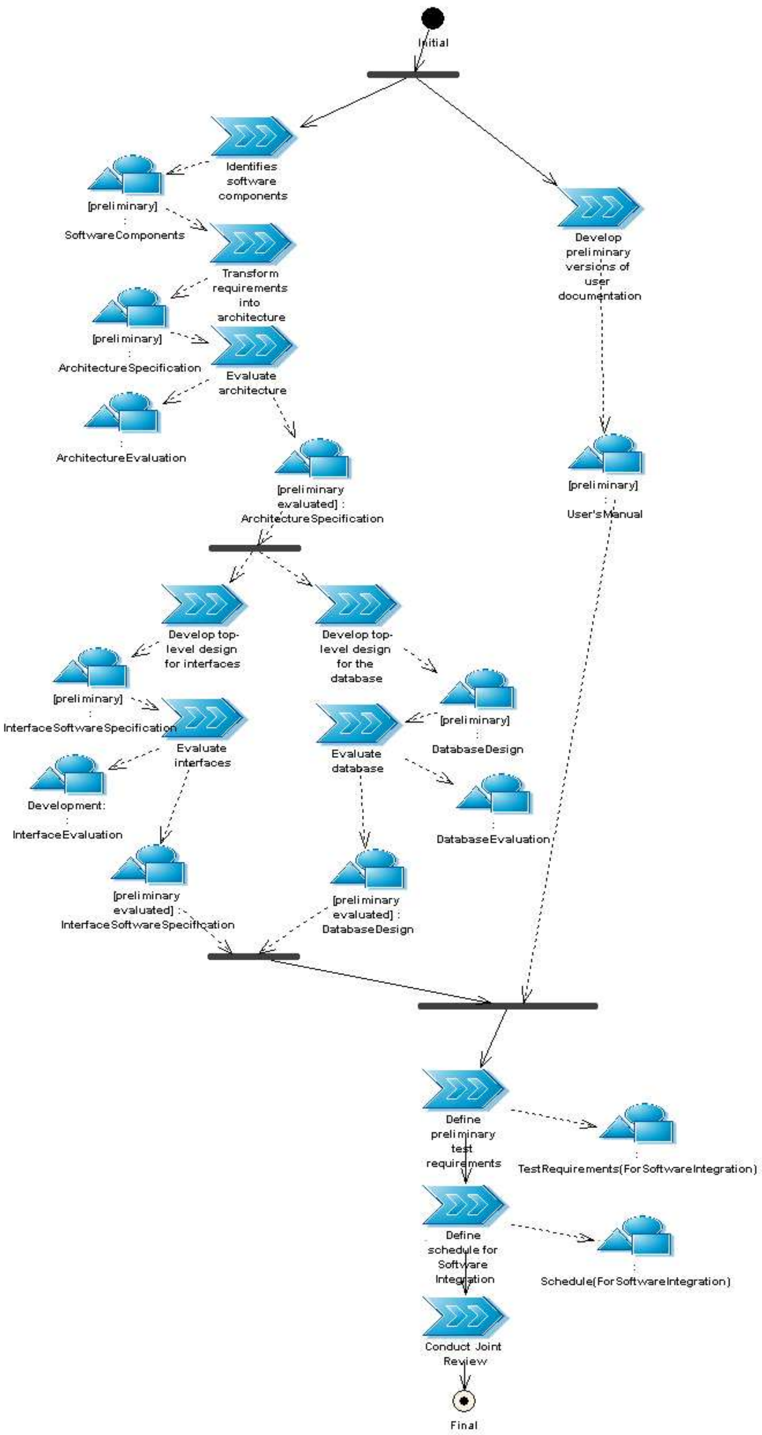

Initial
Identifies software components
[preliminary] : SoftwareComponents
Transform requirements into architecture
[preliminary] : ArchitectureSpecification
Evaluate architecture
: ArchitectureEvaluation
[preliminary evaluated] : ArchitectureSpecification
Develop preliminary versions of user documentation
[preliminary] : User's Manual
Develop top-level design for interfaces
[preliminary] : InterfaceSoftwareSpecification
Evaluate interfaces
Development: : InterfaceEvaluation
[preliminary evaluated] : InterfaceSoftwareSpecification
Develop top-level design for the database
[preliminary] : DatabaseDesign
Evaluate database
: DatabaseEvaluation
[preliminary evaluated] : DatabaseDesign
Define preliminary test requirements
: TestRequirements(ForSoftwareIntegration)
Define schedule for Software Integration
: Schedule(ForSoftwareIntegration)
Conduct Joint Review
Final

Fig. 2-18 Software Architectural Design Phase Activity Diagram

Second problem is how to present reviews, which occurrence don’t depend on finishing deliverables, only are cyclic, or are evoked by other events, such as status review meetings (Performance Reporting process) or quality audits (Quality Assurance process).

Third problem is how to present outputs from project management processes, which have influence on development processes.

Solutions for these problems require extending SPEM metamodel and changing SPEM/UML profile. First solution is presented at bellow process pattern:

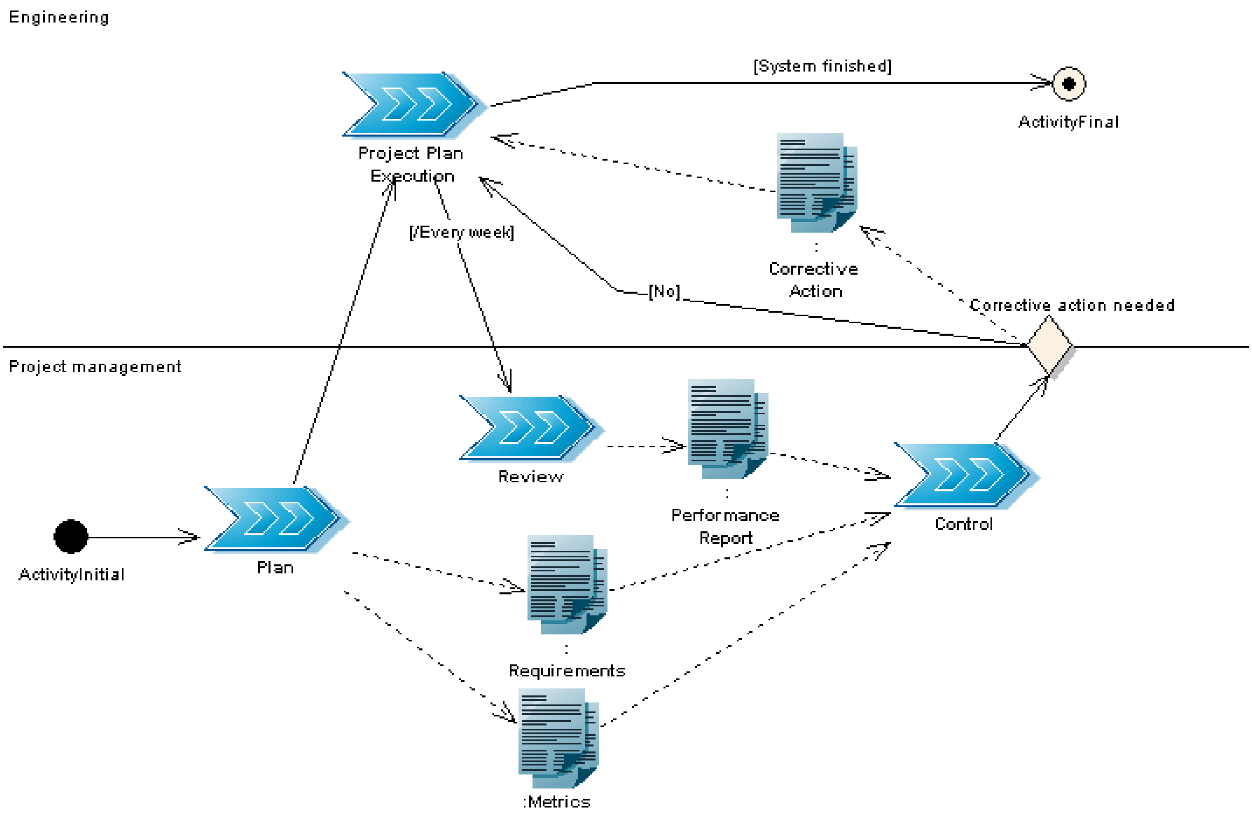


Fig. 2-19 Cyclic review process first pattern

This solution in non compliant to both, SPEM as metamodel and SPEM as UML profile. Semantics of UML Activity Diagram, on which also SPEM elements are presented says, that transition from one activity to another is done, when first activity is finished, and guard condition is fulfilled.

Here are my theoretic considerations, how this model could be read:

Project Plan Execution symbolizes development process, where all development activities take place. From Project Plan Execution to Review activity lead transition that is triggered every week, what means, that every week from development processes control is passed to Review activity, just as on State Diagram. So it is my proposition to present Activity SPEM element as State UML stereotype, and to present project management activities on State Diagram. (In such case Step SPEM model would need to be also modeled as State UML stereotype, and Activity decomposition on Steps would be presented as sub state machine)

Second proposition is extending SPEM metamodel with Frequency element, which would indicate how often Activity would perform within Phase, Iteration or WorkDefinition. Frequency element could be assigned to WorkDefinition element (because all previously listed elements inherit from WorkDefinition) as a TagDefinition. It could take values as for example “Every week”, “Every day” or “As required” what would define, that Activity is executed, when occur such need, or event. Bellow is presented second model pattern. (I present Frequency TagDefinition as Note, because of presentation convenience)

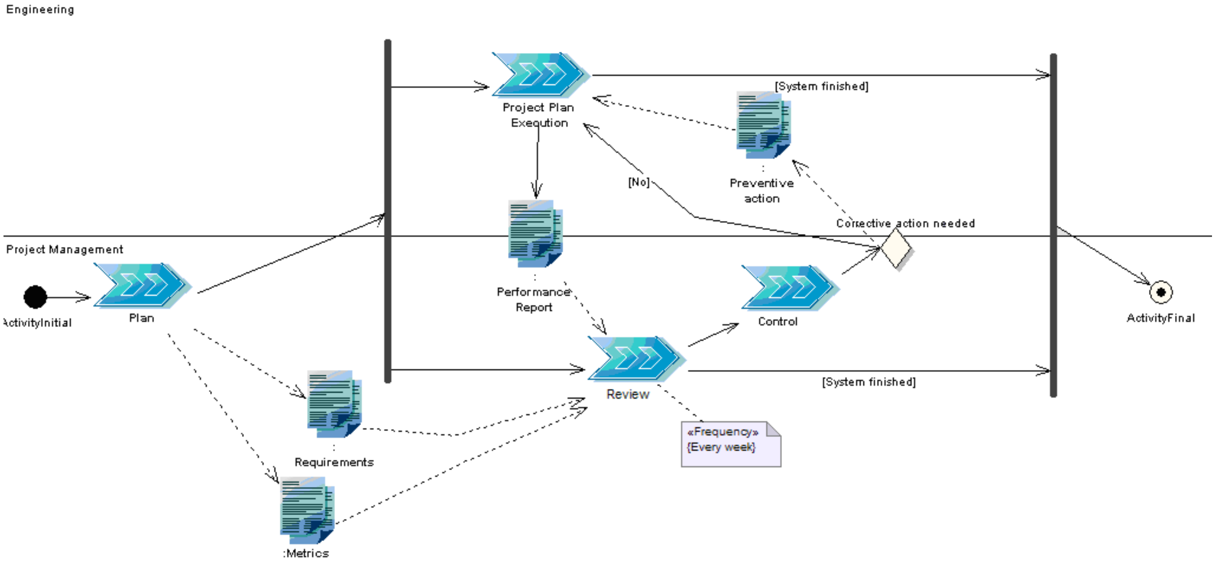


Fig. 2-20 Cyclic review second process pattern

Second model pattern would mean the same what first – Project plan execution would mean execution of development activities. Review activity which is modeled as parallel activity would perform every week, according to new Frequency model element. In my opinion first model pattern would be much more appropriate, and UML compliant.

Control activity on both models is presented only to mark, that after review activity occurs project management process, that control review data. Here is presented proposed solution of third problem – how to pass on information development process. I propose to do this through Document WorkProduct. This stays in harmony with what PMBOK proposes, and it proposes passing through "documented recommendations required to bring expected future project performance into conformance with the project management plan" in case of corrective action, "documented recommendations that reduce probability of negative consequences associated with project risk" in case of preventive action, "documented, authorized directions to expand or contract project scope." (Change request can also change project management plan, procedures, cost or budgets, schedules).

There is one more problem connected with performance systems. Such system in case it would be automatic, should run all the time, collect project related data and write them to some records. Project manager on the base of such records could perform control activity any time. This problem can be reduced to modeling two threads, where one write data and second read data. This solution would also required introduction of new WorkProduct kind - <<record>>. (Such element could be of useful anyway).

Discussion on solutions

I think that both solutions could be applied, although in different occasions.

First proposition – using StateMachine to model project management activities instead of Activity Diagram would be useful, when process engineer would like to present more detailed only project management activities, and events, which evoke these activities. One state called (according to PMBOK 1.0) Project Plan Execution would represent all development processes. Transitions from this state would be triggered by events (according to StateMachine metamodel from UML 1.4), for example time events (after one week), or events which occurrence time we can't define (change request, risk occurrence). This solution could also be applied for activities such as Scope Verification. In this case event, that would trigger transition to Scope Verification could be for example reaching milestone, or finishing deliverable.

Second proposition – using "Frequency" TagDefinition would be helpful by presenting development activities on Activity Diagram together with management activities. This would be useful to present, that some management activities like cyclic reviews occur during project life cycle.

2.5 Unification of project management ProcessComponents with software engineering process

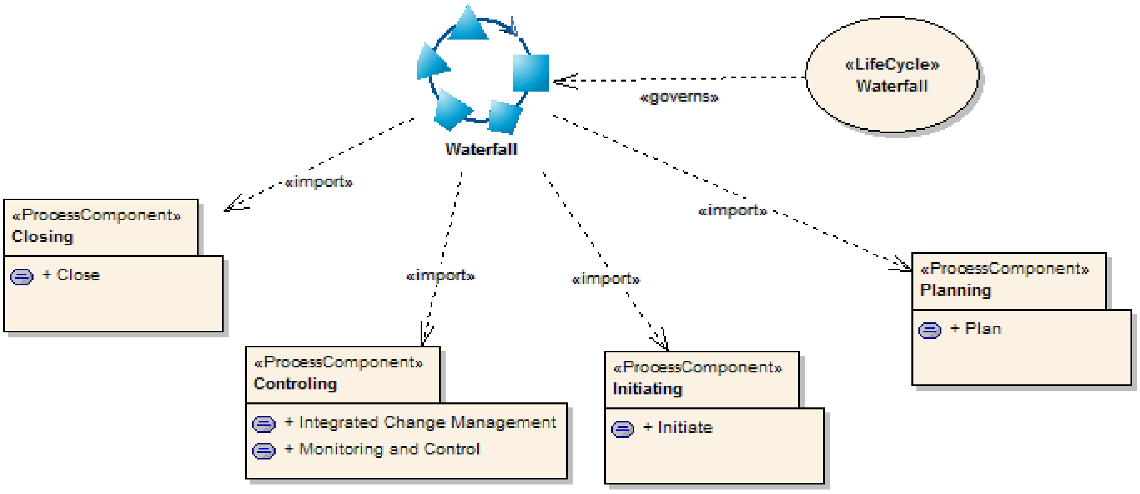


Fig. 2-21 Import of Project Management ProcessComponents by Waterfall Model

Static model that is result of unification process is presented above. As example I have taken Waterfall software development process model, which is symbolized by SPEM Process element, and which is governed by Waterfall LifeCycle SPEM element

Now WorkDefinitions within ProcessComponents should be connected with waterfall development WorkDefinitions. I present one more time decomposition of Waterfall model on phases:

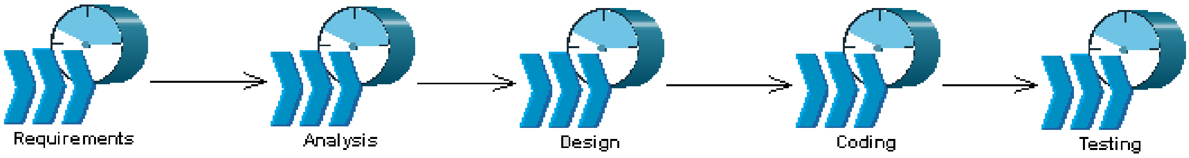


Fig. 2-22 Waterfall model modeled as sequence of phases

Phases of waterfall model contain WorkDefinitions that indicate what to do in current phase. For example Design phase contain Software architectural design and Software detailed design WorkDefinitions.

The best way to unify PMBOK Initiating and Planning ProcessComponents into Waterfall model is creating Strategic Phase at the beginning of life cycle, where this processes could take place. Such approach proposed Jaszkiewicz in book […]

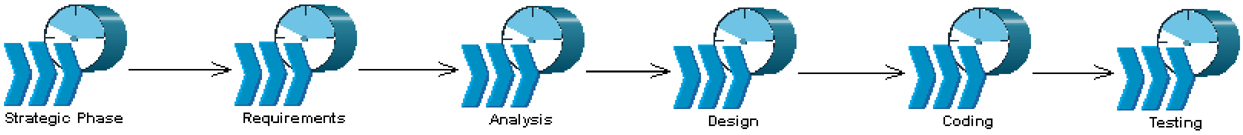


Fig. 2-23 Waterfall model with strategic phase

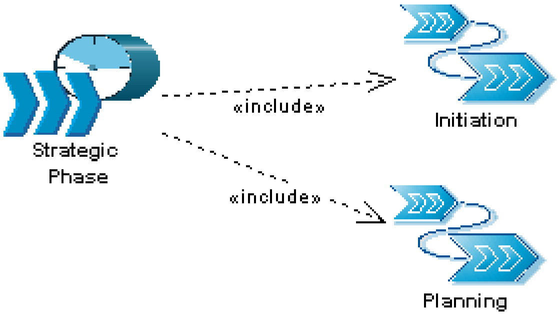


Fig. 2-24 Decomposition of Strategic phase

It would be also reasonable to create Closing phase at the end of life cycle that would contain Closing ProcessComponent. Other phases should incorporate Controlling ProcessComponent.

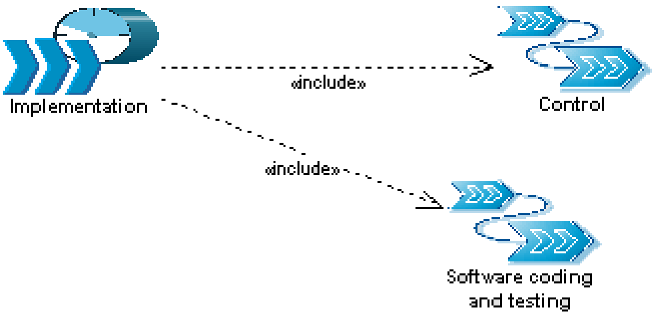


Fig. 2-25 Example of Implementation phase, that contain Control WorkDefinition from Controlling ProcessComponent

We assume, that In implementation phase while Software coding and testing, every week will be conducted review meetings, and control performance. It is reasonable to use one of previously proposed process pattern, to create Activity Diagram that would describe behavior of new model. Model present unification of engineering and management ProcessComponents. There must exists outputs from Software coding and testing in form of Performance Informations and intermediate Deliverables that could serve as inputs to Monitoring and Control WorkDefinition from Monitoring and Control ProcessComponent. Similarly with other WorkDefinitions from other ProcessComponents. (Model don't present all possible outputs from Monitoring and Control and Change Control processes)

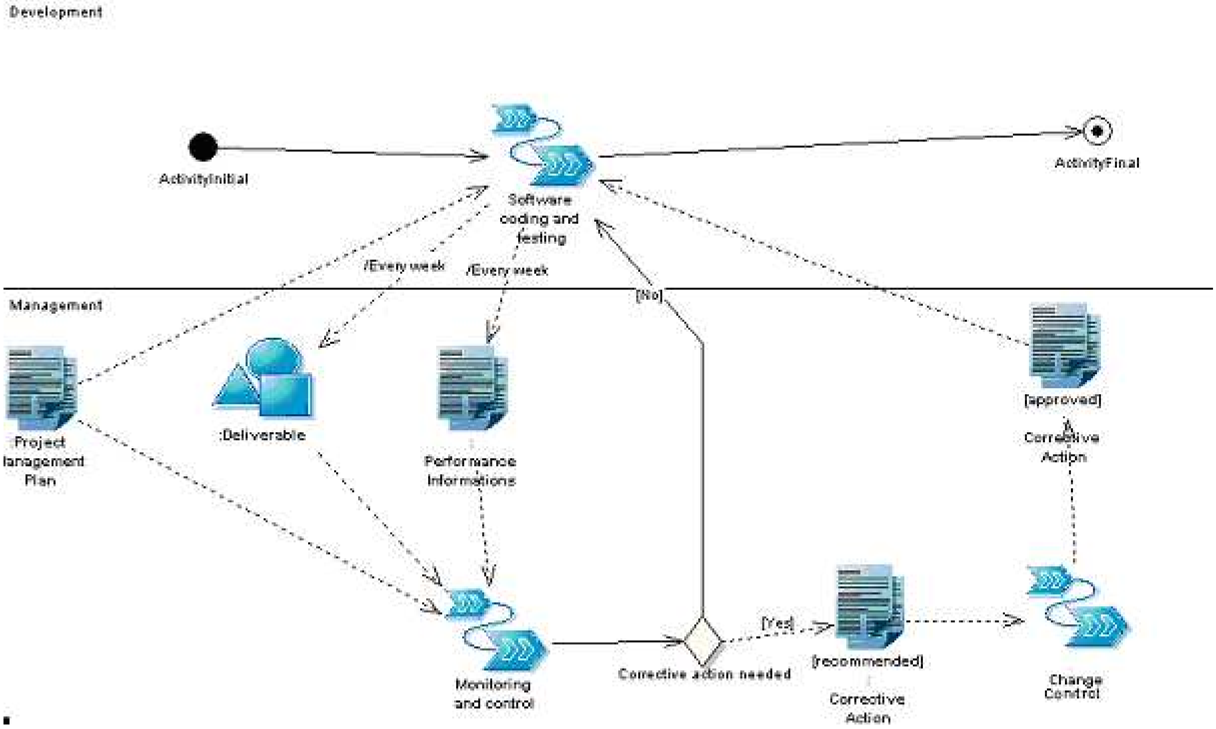


Fig. 2-26 Activity Diagram, that present unification of Project Management and Engineering processes

Second model present little bit different model. In second case when phase is finished Scope of finished Deliverable is verified. In this case Closing ProcessComponent is unified with development process component.

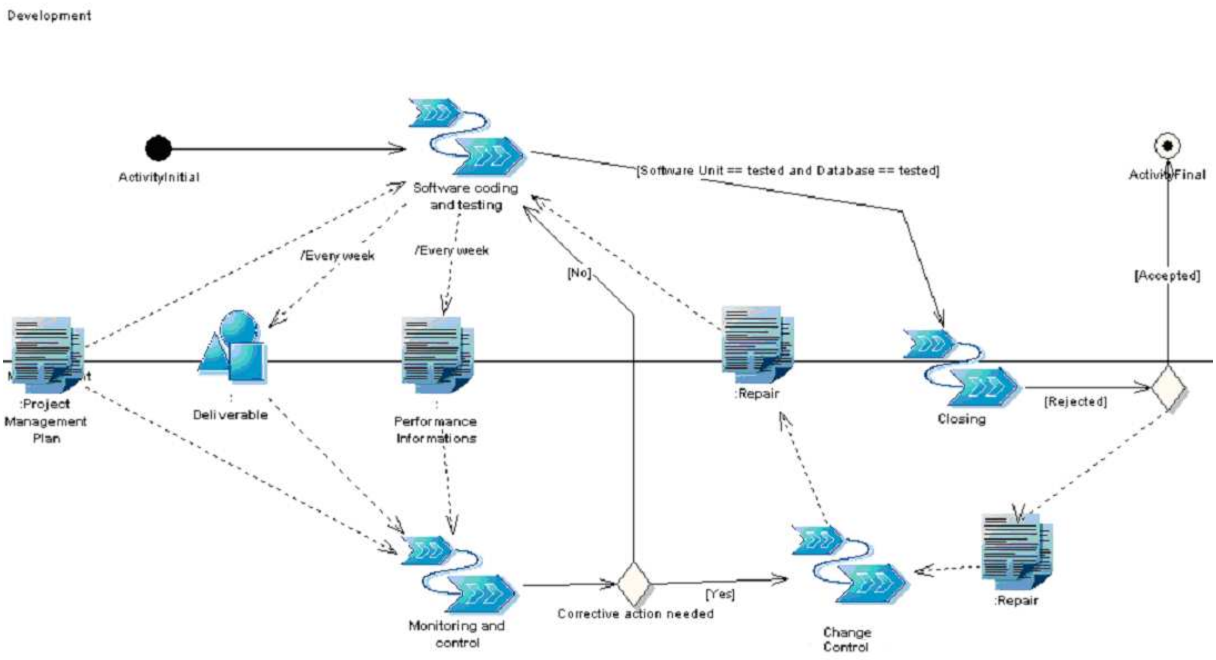


Fig. 2-27 Activity Diagram for unification process – Scope Verification example.

3 Project methodologies

3.1 Introduction

Methodologies have big influence on project management. According to assumed at the beginning of the work definition, methodologies usually have established life cycle, processes, best practices etc. that lead project to successful end. All this parts of methodologies has big influence on project management processes. For example life cycle establishes frame for development work in form of phases. End of phases – so called milestones – are natural decision and control points. Some management processes such as planning take place at the beginning of phase, while other like control spans the whole phase. Processes and best practices of methodologies can also have big influence on project management. Methodologies could require for example that certain management processes are in place.

On the other side, project management processes, for example risk management, can have influence on choosing methodology, which would meet certain assumptions.

Further in this chapter I would like to present several project methodologies, classify them according to several criterion, define which project management and engineering processes occur in which methodology (and relate them to PMBOK processes), and finally present their model in SPEM.

3.2 Methodologies categorization

Jason Charvat in [1] distinguishes between two kinds of methodologies:

1. Project management methodologies (this lays the high-level project framework).
2. Development methodologies (this provides the detail on system design and development).

This stays in agreement with what states book [3]:

“Generally speaking, methodologies have, in the past, been split into two camps — development and management. This means that we have one process model for doing the development work that leads to the final product, and another for managing the development process. Clearly, this is an inefficient way to develop any product, and the Spiral model is currently the only alternative that attempts to roll the software creation and management processes together.”

I’ll describe further two project management methodologies:

1. PRINCE 2
2. Rational Unified Process

And three development methodologies:

1.1 Waterfall
1.2 SCRUM
1.3 XP

There is discussion, how RUP should be treated. J. Charvat classifies RUP as project management methodology. In my opinion it is development methodology, which contains project management discipline, and can be used to lead the project.

These methodologies can be grouped within several criterions important from project management point of view:

1. Software development life cycle (based on life cycles described in ISO/IEC 12207 standard)
2. Are requirements known up front?
3. Methodology contain description of project management methods (Yes, No)

Table x. Comparison of methodologies

Methodology	Life cycle	Are requirements known up front?	Project management
Waterfall	Waterfall	Yes	No
RUP	Iterative	Yes	Yes
SCRUM	Evolutionary	No	Yes
XP	Evolutionary	No	No

1.1.1.1 From above table I have excluded PRINCE 2 methodology, as pure management methodology.

3.2.1 Criterions description

Life cycle criterion

ISO/IEC 12207 characterizes 3 kind of fundamental life cycles: Waterfall, Iterative and Evolutionary. There are several differences between them:

- In waterfall model phases occur in sequence and are done once. Requirements are known at the beginning, and the whole set of requirements is converted into system. Waterfall model can be treated as fundamental life cycle or methodology.
- In iterative model requirements are also known up front, but system is prepared gradually. There is chosen one set from requirements and this set is implemented during first build. In next build next requirements are implemented and added to the first build. And so on until system is ready.

- In evolutionary model requirements are not known at the beginning. System is prepared similarly as in iterative model in builds, and after each build customer adds new requirements. This sequence is repeated until system is ready.

Are requirements known up front?

- In waterfall model requirements must be known up front.

- In RUP requirements are known up front, and system is built in phases, and iterations in which software development activities run parallel.

- XP and SCRUM assumes that requirements are not known up front. Usually only part of them is known. They assume that they can easy be changed, and new requirements can be added, that’s why I categorize them into evolutionary life cycle.

Project management criterion

Some development process models contain description of management processes that complement description of software engineering processes.

3.3 Description of project management methodologies

3.3.1 Rational Unified Process framework

Description

Max Wideman in review of Chartvat’s book argue with this list, and states, that Rational Unified Process and System Development Life Cycle are in fact software development methodologies (thy are in fact software development methodologies, but they also contain several processes for project management, especially RUP contains the whole project management discipline, where are described project management activities). RUP is rate among agile methodologies. There are four main phases in RUP – inception, which (similarly as in SPEM) contains activities that have common theme. Activities within disciplines run parallel.

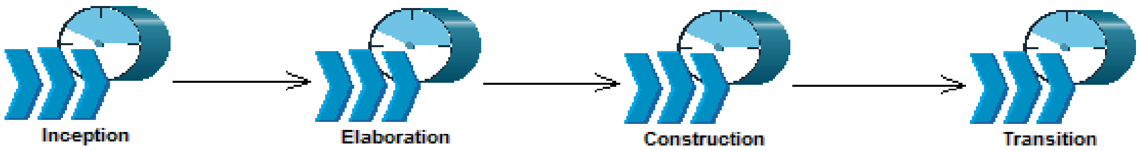


Fig. 3-1 RUP phases

RUP methodology and management

Bellow is presented model for management process that take place inside RUP iteration.

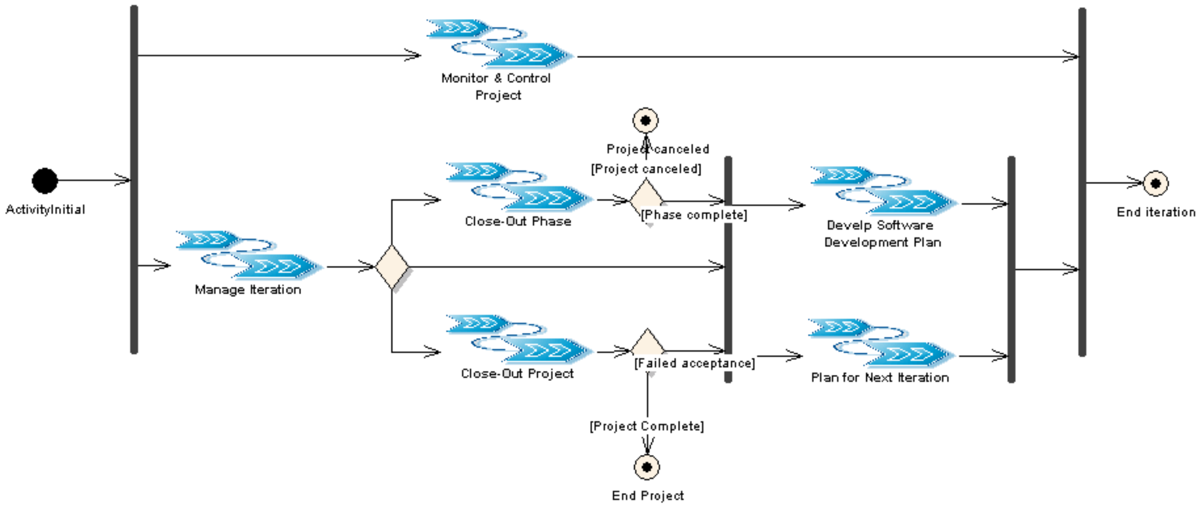


Fig. 3-2 Example of RUP project management process modeled in SPEM

Bellow model present management activities in initial phase of project:

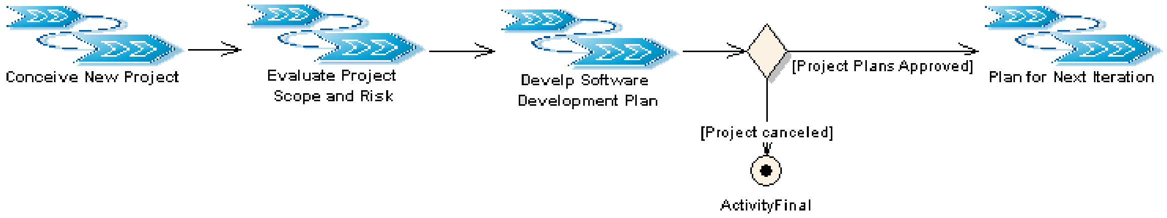


Fig. 3-3 Example of initial phase of project

Bellow model presents more detailed presentation of project management activities within one of RUP iterations inside life cycle. Element on the model are using proposed Frequency TagDefinition. Report Status and Schedule and assign work belongs to Monitor & Control Project WorkDefinition. Acquire Staff, Assess Iteration, Initiate Iteration and Iteration Evaluation Criteria Review belongs to Manage Iteration WorkDefinition. Develop Business Case belongs to Evaluate Project Scope and Risk WorkDefinition (not presented at previous model). Project Plan Execution presents activities from other disciplines, and serves only presentation convenience.

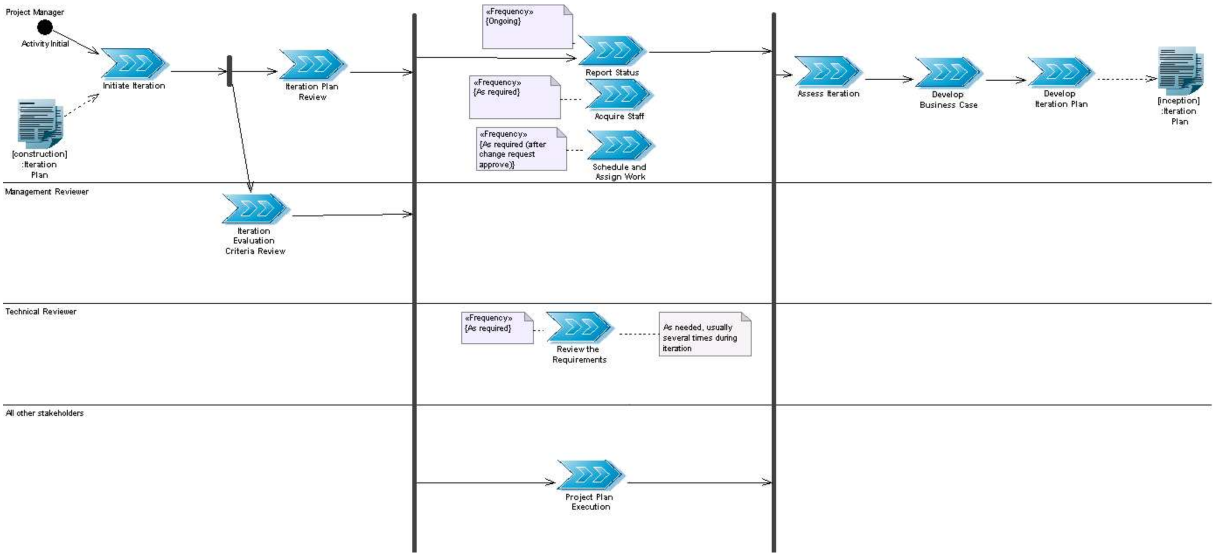


Fig. 3-4 Management activities within RUP iteration

At above model occurs Schedule and Assign Work Activity, which is result of RUP Change Management process. Model of this process from project manager point of view can be presented in SPEM at bellow models:

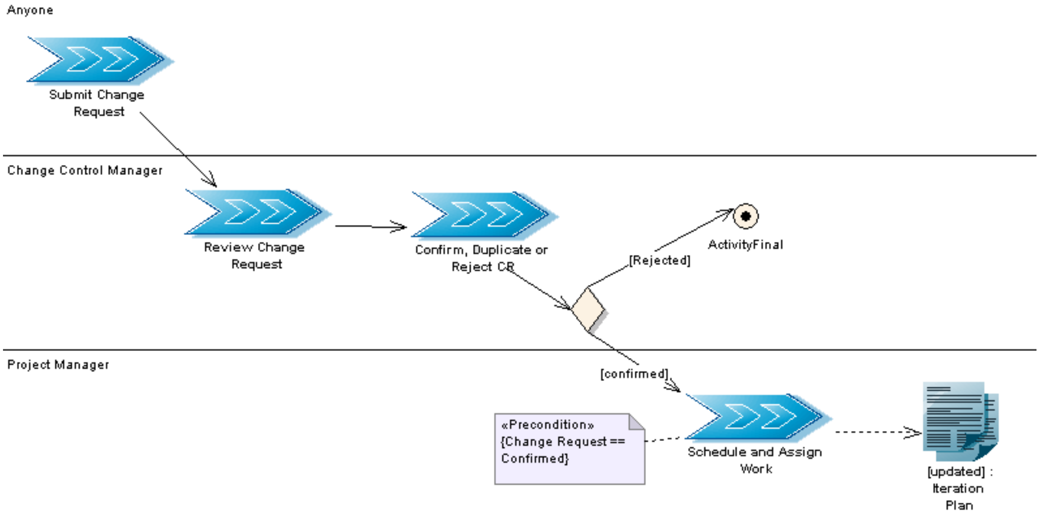


Fig. 3-5 RUP Change Management process

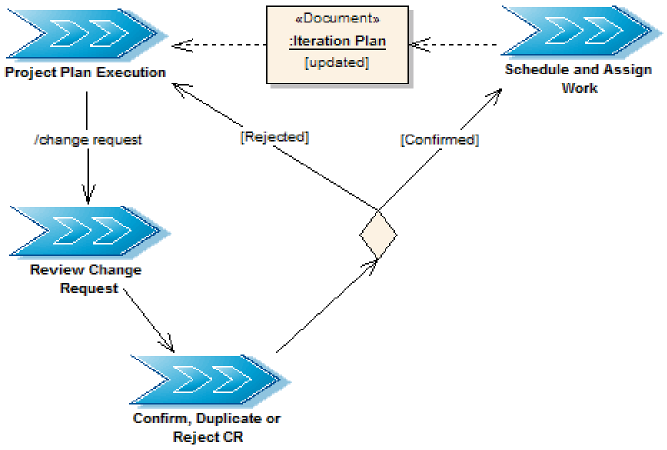


Fig. 3-6 RUP Change management (modeled as StateMachine)

This two models presents two possibilities, on how change request can be modeled. Second one is model created according to proposed previously solution. The whole Change management RUP process can be presented at bellow figure, and can be easy mapped to SPEM:

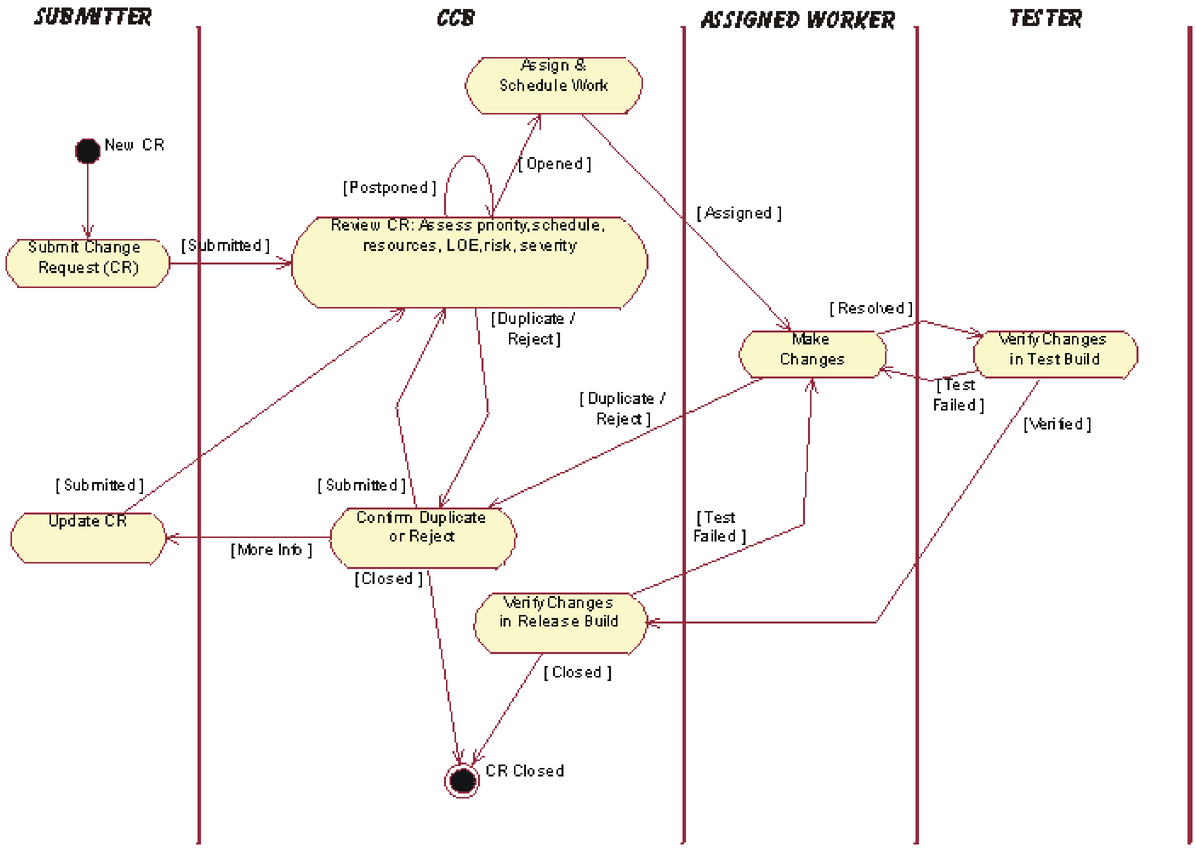


Fig. 3-7 RUP Change Request Management process [source – RUP]

How RUP relates to PMBOK

Detailed mapping of PMBOK to RUP is presented in appendix A. (This is mapping of PMBOK 1.0 to RUP)

Generally PMBOK activities map to RUP project management activities with several exceptions. Bellow list present most interesting cases:

- Project Scope PMBOK processes map to RUP requirements discipline. That is why Scope Verification PMBOK process map not only to Lifecycle Milestone Review from Project Management discipline, but also to Review Requirements activity from Requirements discipline. Generally PMBOK Scope management should map to requirements phase in every methodology, because this phase define scope. In RUP activities from requirements discipline last for all project time.
- Every RUP planning activities map to planning PMBOK processes.
- Every PMBOK control activities map to RUP Report Status and Assess Iteration Activities, and to Configuration and Change Management discipline.
- All PMBOK Time Management processes map to RUP Plan Phases and Iterations and Develop Iteration Plan activities except from Schedule Control process which case is describe above.
- RUP don't have activities that would relate to Project Cost Management PMBOK processes. It is only taken into consideration in Planning for Phases and Iterations and Staff Acquiring.
- Quality Planning has in RUP taken into consideration within Develop Quality Assurance Plan, Define Monitoring and Control Process, and Develop Measurement Plan activities. Quality Assurance is mapped to Configuration & Change Management discipline, similarly as Quality Control, but to Quality Control process are rated every Test and Review Activities across all Disciplines.
- Monitoring and Control RUP activity occur by the way Project Plan Execution PMBOK process and Performance Reporting process.
- Risk in RUP is directly mapped to Develop Risk Management Plan and Identify and Assess Risks by the way of PMBOK planning process. Risk Control responds to Configuration & Change Management.

3.3.2 PRINCE2 framework

Prince 2 is pure management methodology, which means, that it doesn't take into consideration development processes. PRINCE 2 methodology can be applied to every kind of project, no matter which methodology it uses. It is process-orientated approach, and is decomposed on phases:

- Starting up a Project
- Initiating Project
- Managing Stage Boundaries
- Controlling a Stage
- Managing product delivery
- Closing a Project

Influence on all this phases has Planning process. It considers key project processes:

- Planning an initiation phase
- Planning a project
- Planning a phase
- Producing an exception plan

Supervision over all phases and processes has Direction a Project process. “This process is aimed at the project board, who manages by exception, monitors via reports, and controls through a number of decision points.”

To summarize, Starting up a Project and Initiating Project are phases, which cause project start. Managing Stage Boundaries is phase (process) which takes place at the end of development phases or iterations. Project manager deliver to project board appropriate information which have influence on decisions at crucial points. Project board has competition to approve current phase and authorize the start of the next phase.
Controlling a Stage is phase (process) in which project manager:

- authorize work to be done
- gather progress status
- watch for changes
- report
- implement necessary corrective actions.

Managing product delivery is phase (process) where project manager supervise the project through

- ensuring, that work conforms to the requirements,
- ensuring that the work is done,
- assessing work progress and forecasts progress, and finally
- through obtaining approval for completed products.

Managing product delivery and Controlling Stage are tightly coupled.
Closing project is process that takes place at the end of project, and finishes it.

How PRINCE 2 map to PMBOK
Conception of PRINCE 2 an PMBOK is very similar. PRINCE 2 distinguishes processes similar to PMBOK process groups, and place them in similar places of the project. For example Starting up a Project, Initiating Project would respond to Initiating process group, Managing Stage Boundaries and Controlling a Stage would respond to Monitoring and Controlling process group, Managing product delivery respond to Executing process group and Closing a project respond to Closing process group. Generally PRINCE 2 models would look very similar to PMBOK models.

3.4 Description of Software Development Methodologies

3.4.1 Waterfall

Description
The waterfall model is a software development model (a process for the creation of software) in which development is seen as flowing steadily downwards (like a waterfall) through the phases of requirements analysis, design, implementation, testing (validation), integration and maintenance. The term was introduced in 1970 by W.W. Royce. [W.W. Royce Waterfall] (It should be emphasized, that Royce was not supporter of strict waterfall model, in fact, he characterizes “do it twice” and principle and allow for returns from further to previous phases.) The phases are parts of a top-to-bottom procedure that sequentially reduces an abstract concept to executable software program. Each phase will be separately examined.
In literature there are many versions of waterfall model. Royce proposed following phases:

1. System requirements
2. Software requirements
3. Analysis

4. Program design
5. Coding
6. Testing
7. Operations

I'll consider only Requirements, Analysis, Design, Coding and Testing phases. It shouldn't have big influence on discussion, but can make diagrams more clear.

It depends on how software engineer will treat the waterfall model, and on how big project is, the above phases can be presented as Activities, WorkDefinitions or Phases. In small student projects above phases could be presented as Activities, while in big projects should rather be modeled as Phases. This is because in small project the phases could mean exactly what is to be done – analyze, design, implement and test program, while in big projects phases can be decomposed on many smaller activities for example Design phase can be composed of database design WorkDefinition, interface design WorkDefinition and system design WorkDefinition.

In my researches I find, that waterfall model describe only software development processes, and all principles considered software development. Bellow figure present sequence of software development phases:

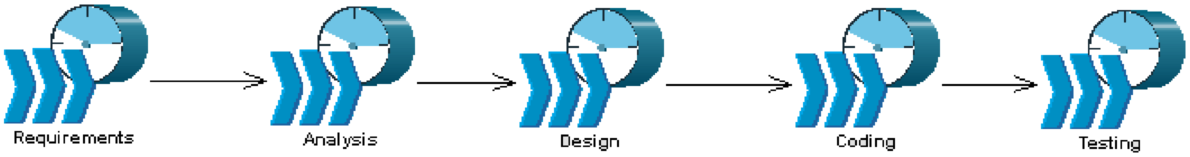


Fig. 3-8 Waterfall model modeled as sequence of phases

Waterfall methodology and management processes

Management processes that should occur in waterfall methodology:

1. Planning processes - project should be planned, actually this is one of most important process in waterfall model – it require big up front planning. If plan don't respond to reality, usually higher management blame project manager for this. Waterfall project are usually used to project with low risk, in areas that are well known, and with highly predictable durations so it is easy to create plan.
2. Project plan should execute through phases.
3. Project should be monitored and controlled. Every phase should end with review of deliverables, and decision if project should go to the next phase.
4. In waterfall model changes are impossible to implement. The whole effort is done to establish stable requirements. So there should not occur processes that concern introducing changes into project. (Scope control in PMBOK). In reality changes can occur, but it is hard to introduce them in waterfall model.

3.4.2 SCRUM

Description

Scrum consists of three phases: Pregame, Development and Postgame.

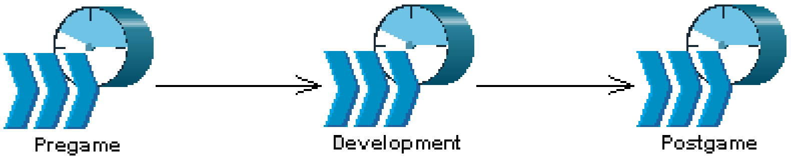


Fig. 3-9 SCRUM phases

During Scrum Planning project team, tools and other resources are defined, risk is assessed, training needs are stated and controlling and verification management approval issues are considered. In planning phase high level architecture is prepared on the base of current Backlog list. This architecture is reviewed. Also plans for releases are prepared.

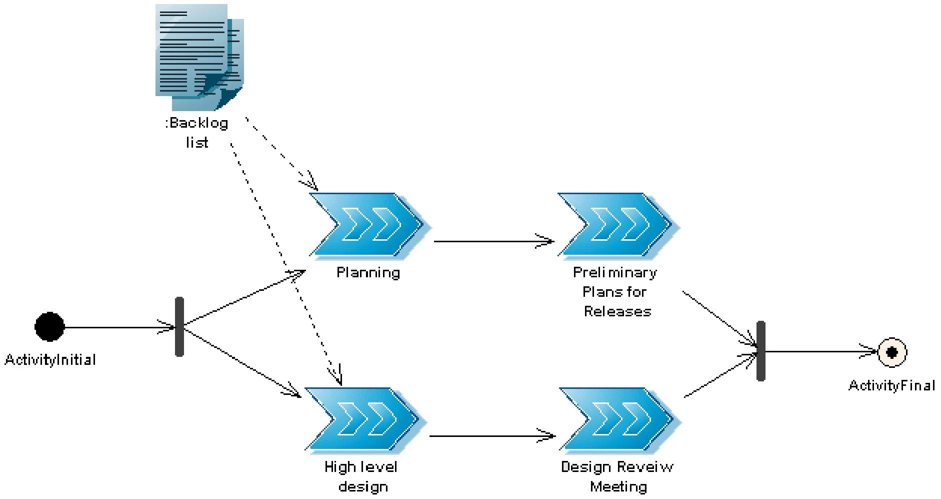


Fig. 3-10 SCRUM Pregame phase

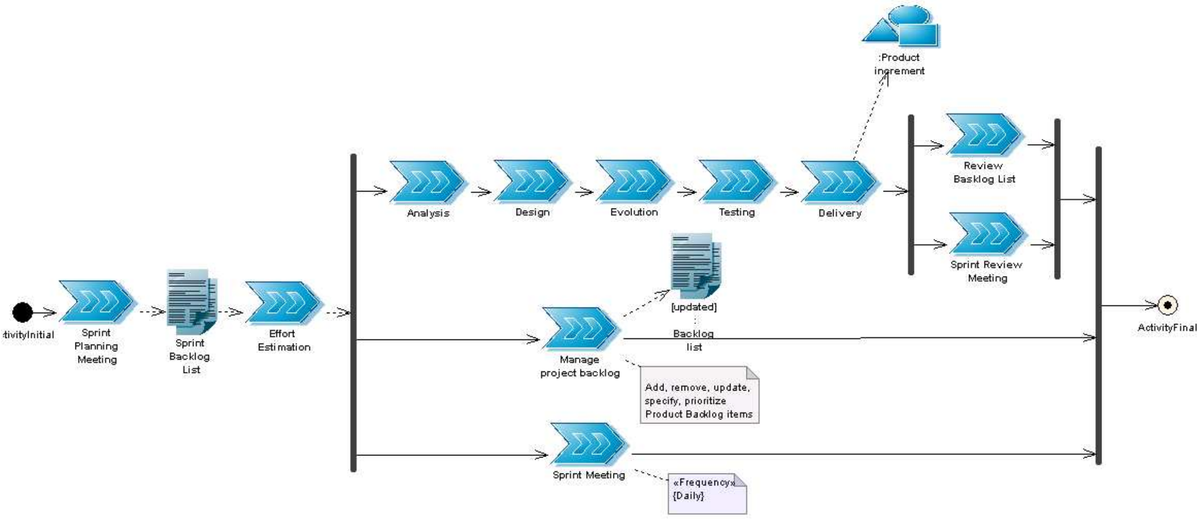


Fig. 3-11 Sprint activity diagram

The Development phase is agile part of methodology. It consists of so called sprints, during which traditional elements of software development process are performed. There are as many sprints as needed. “At every iteration, the updated product Backlog is reviewed by the Scrum Team so as to gain their commitment for the next iteration.” [Agility]

“The different variables (time frame, quality, requirements, resources, implementation technologies and tools, etc.) are observed and controlled through various Scrum practices during the Sprints […] Rather then taking these matters into consideration only at the beginning of the software development project, Scrum aims at controlling them constantly in order to be able to flexible adapt to the changes.” [Agility] Post game phase contains closure of the release. If all requirements are complete, and new requirements are not added to Backlog list system can go through integration, system testing and documentation.

SCRUM methodology and management

In agile methodologies should occur every from project management processes, but differently than in more traditional approaches. There is no big up front planning. General project plan is crated at the beginning of project, but detailed project plan is performed only for next iteration. As iterations are very short, there is also not as big need for risk management as for classic project models, because risk is highly managed through short iterations – in case of unexpected event occurrence it is taken into considerations in plans for next iterations. In reality certain amount of risk planning is desirable.

Every day 15 min. meeting take place where problems are discussed, actions for management are defined, and status of completed items to date is defined.

At the end of every Sprint review meeting takes place. During this meetings project manager inspects results, assess changes in the business environment, and determines what to do next.

3.4.3 XP

Description

In XP occur several phases. In Exploration phase Customer writes out the story cards he wishes to be included into first release. During this time team familiarize themselves with tools and technique. In Planning phase there is set priority order for the stories, and stories are chosen. Schedule is developed for first iteration. In Iterations to release are several iterations that are performed, before first release of system is prepared. In first iteration general architecture for system is developed. Customer select stories for each iteration. Customer functional tests are run at the end of each iteration. At the end of last iteration system is ready for production.

In Productionizing phase extra testing is done to check system performance. At this stage new changes still can be made. After Productionizin phase system is running for customer, and new iterations are still done. This requires efforts both for development and customer support. This phase is called Maintenance phase. This phase may require incorporating new people into team and changing team structure. In Death phase customer doesn’t have new stories to be implemented. In this phase finally documentation is written, as there are no more changes to architecture, design and code. Below model presents XP phases.

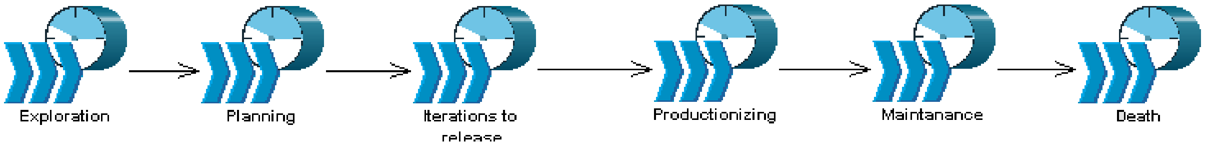


Fig. 3-12 XP phases

Every phase consists of several iterations. It depends on how big system user needs, and it's impossible to plan for iterations. Detailed description is needed for Planning phase – this phase can occur at the beginning of project, to characterize general project plans, but planning activities (same as for Planning phase) occur also before every iteration. Similarly with Exploration phase – gathering user stories occurs at the beginning of the project, but is also occurs at the beginning of every iteration – stories are updated and new are added then.

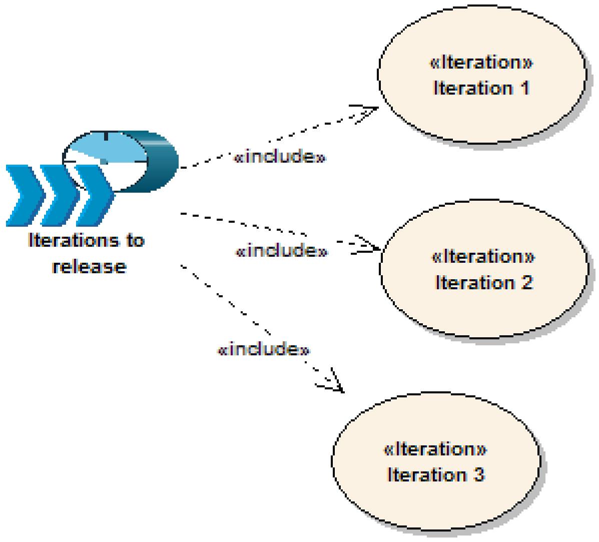


Fig. 3-13 XP Phases consists of several iterations

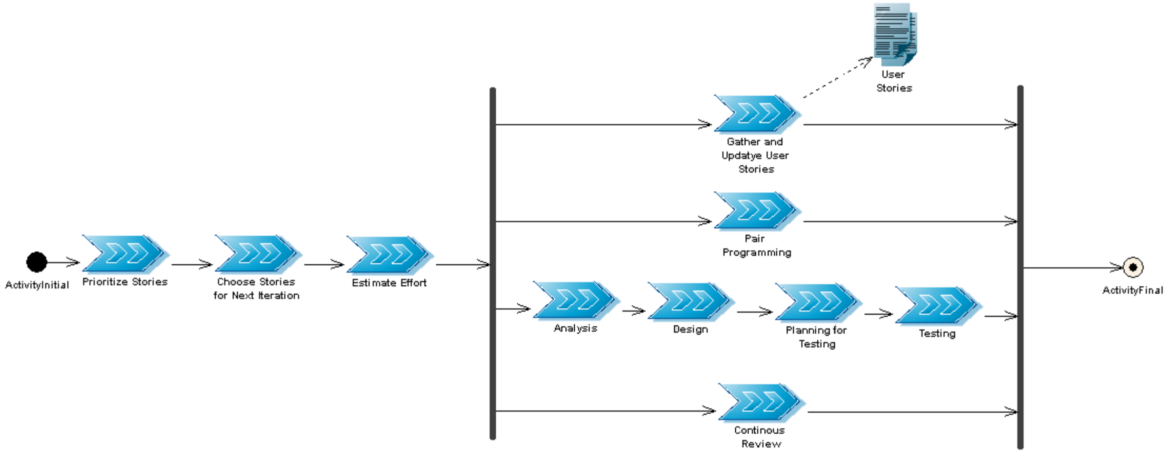


Fig. 3-14 XP Iteration from Iterations to release model

XP methodology and management

Although XP doesn't formally have project management elements, in fact in several aspects can be mapped to PMBOK. For example choosing stories to implement for every iteration could be mapped to Scope Management PMBOK process area. Similarly Effort Estimating for this stories could map to Project Time Management. Several XP practices as pair programming or TDD is kind of Quality Assurance. Finally functional test performed by client can be mapped to Scope Verification.

3.5 Description of Frameworks

3.5.1 ISO/IEC 12207

ISO/IEC 12207 “is a common framework for software life cycle processes”. It contains description of all processes needed to develop software. Processes presented there can be tailored as needed. Standard group processes within three process groups – primary, supporting and organizational life cycle processes. Example of waterfall model that is build according to ISO/IEC 12207 framework is available in appendix D.

How ISO/IEC 12207 relates to PMBOK

ISO/IEC 12207 standard distinguish management processes as belonging to organizational life cycle processes.

Project management process contains following activities:

1. Initiation and scope definition
2. Planning
3. Execution and control
4. Review and evaluation
5. Closure

Which actually respond to PMBOK process groups. (In PMBOK execution and control are separate process groups, but review and evaluation is included into control process group).

4 Transformation of SPEM to MS Project

The objective of program that transforms SPEM models to MS Project models is to facilitate project manager work. Project manager having process models in SPEM will be able to easily and fast create MS Project schedule on the basis of template that will be result of transformation. Bellow figures present examples of source and outcome of transformation:

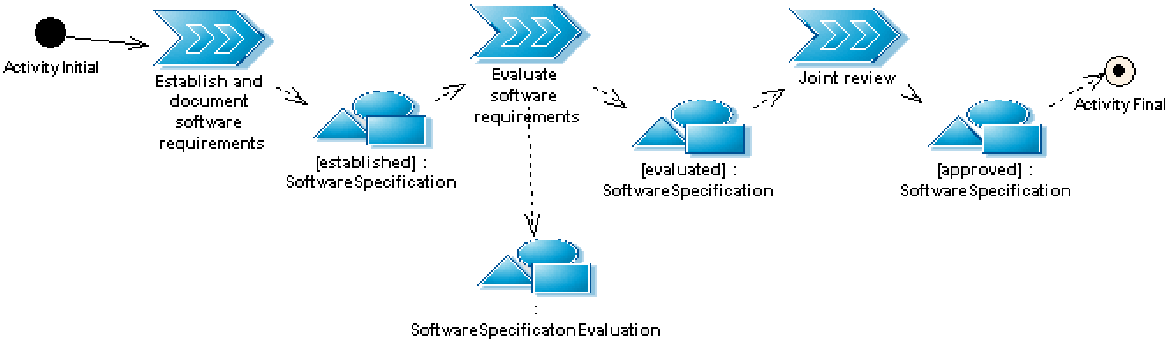
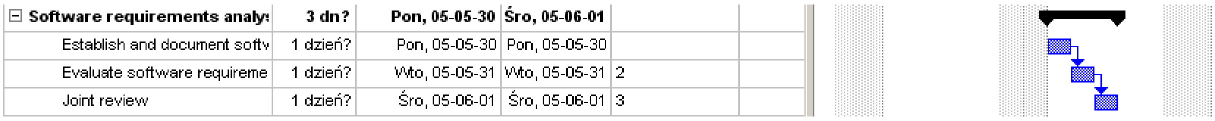

Activity Initial
Establish and document software requirements
[established] : SoftwareSpecification
Evaluate software requirements
: SoftwareSpecificatonEvaluation
[evaluated] : SoftwareSpecification
Joint review
[approved] : SoftwareSpecification
Activity Final

Fig. 4-1 Source for transformation (Activity Diagram modeled in SPEM – Enterprise Architect tool)



⊟ **Software requirements analy:**	**3 dn?**	**Pon, 05-05-30**	**Śro, 05-06-01**		
Establish and document softv	1 dzień?	Pon, 05-05-30	Pon, 05-05-30		
Evaluate software requireme	1 dzień?	Wto, 05-05-31	Wto, 05-05-31	2	
Joint review	1 dzień?	Śro, 05-06-01	Śro, 05-06-01	3	

Fig. 4-2 Result of transformation (Gantt chart modeled in MS Project)

Application is build as add-in to MS Project program. It has its own button in MP Project toolbar.

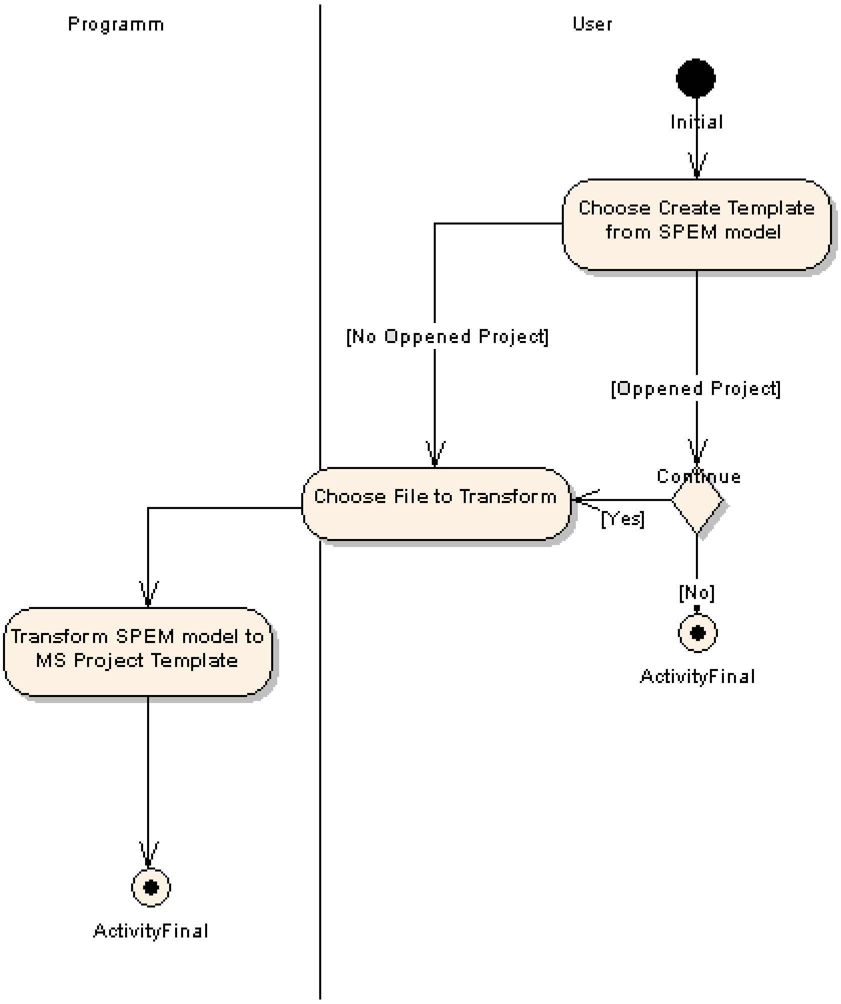


Fig. 4-3 Interaction with user interface

Interaction with interface is very simple:

1. User chooses to create template from SPEM model – user click add-in button.
2. There occurs Open File window, where user can choose xml file in which SPEM process model is saved.
3. User click OK button.
4. If there is opened MS project, program ask user if he want to continue, and close current project
5. User choose not to continue – program comes back to previous state
6. User chooses to continue – program create template on the base of SPEM model.

The whole logic of transformation is saved in external xslt files. The whole schema of Enterprise Architect file xml schema that contains SPEM model, MS Project schema that is generated and the xslt transformation that transforms EA files into MS Project files is available in Appendix B.

EA Project files elements:

To properly read EA projects, elements need to be read from <UML:Diagram > element, where

```
<UML:ModelElement.taggedValue>
        <UML:TaggedValue tag="parent" value="EAID_253A488A_166A_46b1_B253_4121696E83A3"/>
</UML:ModelElement.taggedValue>
```

xml element defines element which is represented through diagram element, for example

```
<UML:SimpleState name="Requirements" xmi.id="EAID_253A488A_166A_46b1_B253_4121696E83A3">
```

Transitions between model elements are signed as <UML:Transition> element, whith source and target elements:

```
<UML:Transition xmi.id="EAID_0CCB54A9_0C77_4e62_A799_E9B7B1EC5CDC" visibility="public"
source="EAID_373F4651_9678_4432_A3AC_5161F75DD98C"
target="EAID_FC2D0A1D_A489_4323_BD12_124F8BC2C403">
```

MS Project files elements:

To create MS Project Gantt chart, at least following elements need to be written in XML source file [Description on the base of MS Project help]:

1. **Task** - define Task element, which respond one task on Gantt chart
2. **UID** - global unique task id
3. **ID** – unique task id
4. **Name** - name
5. **Type** - In a <Task>, <Type> is the type of task (fixed units, fixed duration, or fixed work).In a <PredecessorLink>, it is the type of task link (FF, FS, SF or SS).
6. **WBS** - A unique code (work breakdown structure) used to represent a task's position within the hierarchical structure of the project.
7. **OutlineNumber** - Indicates the exact position of a task in the outline. For example, "7.2" indicates that a task is the second subtask under the seventh top-level summary task. The <Outline Number> is a string indicating the task's position in the outline hierarchy; "5.2.6" is an example. The <OutlineLevel> is a simple number, such as 2.
8. **OutlineLevel** - The number that indicates the level of a task in the project outline hierarchy. The <OutlineLevel> is a simple number, such as 2. The <OutlineNumber> is a string indicating the task's position in the outline hierarchy, "5.2.6" is an example. (yes)
9. **Recurring** - Indicates whether a task is a recurring task
10. **Milestone** - For a <Task>, <Milestone> indicates whether a task is a milestone. For an <Assignment>, it indicates whether the assignment task is a milestone.
11. **Summary** - Indicates whether a task is a summary task. (yes) I think this indicate if task is WD or Activity in SPEM example.

Example of MS Project XML file schema:

```
<Task>
  <UID>2</UID>
  <ID>2</ID>
  <Name>A1</Name>
  <Type>0</Type>
  <WBS>1.1</WBS>
  <OutlineNumber>1.1</OutlineNumber>
  <OutlineLevel>2</OutlineLevel>
  <Recurring>0</Recurring>
  <Milestone>0</Milestone>
  <Summary>0</Summary>
</Task>
```

5 Summary

SPEM as process modeling language is applicable to many aspects of process engineering. It enables to describe existing processes (descriptive purpose) or designing new processes (prescriptive purpose), In my work I concentrate on prescriptive aspect.

In my work I've presented several elements and capabilities of SPEM. Most of them considered using SPEM to model management processes in context of engineering processes. The main aims of my work were to define if project management processes can be modeled using SPEM profile. In order to answer for this question I needed to identify project management processes.

As source of knowledge from project management area I have used PMBOK. On the base of processes described in PMBOK I needed to define which project management activities occur within project, and in which place of life cycle do they occur. Further section contains summary of this research.

PMBOK described processes in terms of knowledge areas – areas that group processes within common theme. They describe processes in aspects of inputs, outputs and tools and techniques that are used to create outputs from inputs. It is possible for some PMBOK processes – mostly initiating, planning and closing processes to be modeled as SPEM activities. It is not always possible in case of controlling activities.

PMBOK processes that belong to initiating, planning and closing processes groups are mostly sequential, and generally can be modeled as sequential activities (beside complicated planning process). In connection with engineering processes they can be presented as initial or final processes on process model. There is no problem in modeling such processes. It is differently in case of executing and controlling processes. Executing and controlling processes concerns execution of project plan. Executing processes can be regarded as development activity. Controlling process has generally two aims. First role of control processes is to control project in areas of scope, time, costs, quality, human resources and risk. Controls in scope, time and costs can results in change request, or corrective action request, to achieve project performance accordance to established baselines. Corrective actions can very often lead to change request. Controls in quality, risk and human resources can lead similarly as previously to change request or corrective action, but additionally to preventive action, which is action taken to minimize possible risk influence on project, and to ensure highest possible quality. Second role of control processes is to identify and analyze changes, what results in approving or rejecting changes. Change request can be result of control processes, corrective or preventive actions (which are also outputs from control processes), execution of project plan or external factors. Analyzing change request require that same techniques as for controlling processes, and in case of approving changes it require repeating some of planning processes (re-planning). Controlling processes as input require information on performance in above listed areas. PMBOK describes performance reporting from communication knowledge area as process responsible for supplying performance reports to controlling processes. One of its techniques is review activity. But PMBOK describes also review activity as technique of controlling activities. On the base of above descriptions I decided to model communication process as review activity, which assures input information for controlling processes. This approach is similar to the way in which software development methodologies assures control over project. In methodologies review activity is distinguished (it can be called inspection, joint review, evaluation etc.). Generally according to methodologies and PMBOK descriptions, such control activities should occur at the end of phases (in this case they should be connected with scope verification process, and this is the place, where software development life cycle concept occur), or cyclic – for example once a week. Process engineer can present control process, that take into consideration only management part of the process, but if process engineer would like to present engineering and management activities at the same model, then the first case is easy to model while second case is impossible to model. This is because of semantics of UML activity diagrams. SPEM as UML profile uses UML activity diagrams to present process flow. Transition from one to the other activity on activity diagram is possible, when first activity is finished and guard condition is fulfilled. So it is possible to model reviews that occur at the end of phases, when all previous activities are over. But it is impossible to present cyclic review activities.

I proposed two solutions of this problem. The first solution is to present activities on state chart diagram as states. Semantics of state is different than semantics of activities. Transition from one state to the other is possible, when some event occur. In particularly this can be time event. So it would be possible to present activities as states (state would be proxy for SPEM activity element) and model cyclic transitions as time events. In terms of SPEM as states should be modeled WorkDefinitions – all elements that inherit from WorkDefinition (LifeCycle, Phase, and Activity) could be then modeled also as state. Decomposition of WorkDefinition on subWork could be presented as state decomposition on sub states. Similarly decomposition of Activity on Steps would be modeled as state decomposition on sub states (in SPEM as UML profile Steps are modeled as states).

Similar procedure could be proposed to model change request – this would be other kind of event that would evoke project management process concerns change management. This solution is useful, when process engineer would like to present dependencies between engineering and management processes. It allow for presenting different level of details – for example all engineering processes can be presented as one state, and management processes can be presented with intended level of detail.

I would like to add, that state charts diagrams can be used in SPEM, but in context of object states, for example WorkProduct states and events are not allowed.

Second solution is to add Frequency TagDefinition to Activity element. This TagDefinition would indicate how often or when certain activity occurs. Activities should be model as parallel activities – for example coding activity that would related to engineering process, and review activity with Frequency “every week” Tagged Value.
Such solution is betters to present more detailed software engineering activities, and only to mark that management activities occurs.

My work also contains presentation how project management methodologies can be modeled in SPEM.

According to process engineering definitions, SPEM allow for several kinds of modularization. One of elements that support modularization is ProcessComponent. In my work I have presented, how project management processes could be modeled using ProcessComponents, from which project development processes or methodologies can be created in process of unification.

I proposed to distinguish five ProcessComponents – four according to PMBOK process groups – initiating, planning, controlling and closing, and one that is distinguished from controlling process group – change management. I’ve presented how this components can be used to create process model that contain engineering and management processes. Unification process was conducted with waterfall model processes. I think application of such components is very useful in case of process engineering, and has similar role as using ready components in software engineering.

In my work I have also analyzed several methodologies, classified them according to several criterion, define which project management and engineering processes occur in them (and relate them to PMBOK processes), and finally present their models in SPEM. Bellow I summarizes information concerns them.

I have analyzed:

- Management methodologies:
    - RUP
    - PRINCE 2
- Methodologies:
    - Waterfall
    - XP
    - Scrum
- Frameworks:
    - ISO/IEC 12207

RUP is in fact software development methodology, which contains project management discipline. It contains several project management workflows:

1. Conceiving new project
2. Evaluating project scope and risk
3. Developing software development plan
4. Planning for next iteration
5. Managing Iteration
6. Monitoring & Controlling project
7. Close-out

Which workflows also generally responds to PMBOK process groups: Conceiving new project responds to initiating process group, Evaluating project scope and risk, Developing software development plan and Planning for next iteration responds to planning process group, Managing Iteration responds to Executing process group Monitoring & Controlling project responds to Monitoring & Controlling process group and Close-out responds to Closing process group.

PRINCE 2 is pure management methodology. It is process-orientated approach and the methodology is divided into phases. PRINCE 2 phases can be generally easy mapped to PMBOK process groups. For example Starting up a Project, Initiating Project would respond to Initiating process group, Managing Stage Boundaries and Controlling a Stage would respond to Monitoring and Controlling process group, Managing product delivery respond to Executing process group and Closing a project respond to Closing process group. Generally PRINCE 2 models would look very similar to PMBOK models.

Waterfall is in fact process model, which can be applied as methodology. In the form originally presented it didn’t contain project management activities, but in reality it should contain activities from all PMBOK process groups, with emphasis on planning process.

XP is an agile project methodology. It contains several specifics practices. Although it don’t define project management processes, in methodology description occur notions that could be related to PMBOK processes. For example choosing stories to implement could be considered as planning process, while customer acceptance tests at the end of each iteration as scope verification.

SCRUM has more detailed description of management activities than XP, but is also agile methodology. It defines planning activities for all iterations, review meetings that have place every day, and spring review meeting which occur at the end of each iteration.

Generally agile methodologies enable bigger control over project execution, risk and changes through short iterations. They also don't require big up front planning – more appropriate is general project plan and detailed plan for next iteration.

Finally I have analyzed ISO/IEC 12207 standard. ISO/IEC 12207 is a framework. It describe the whole set of processes needed to perform software development project. Processes presented there can be tailored to fit real project needs. This standard contains not only software engineering processes, but also among the others project management, supporting and quality assurance processes. Project management process contains following activities:

6. Initiation and scope definition
7. Planning
8. Execution and control
9. Review and evaluation
10. Closure

Which actually respond to PMBOK process groups. (In PMBOK execution and control are separate process groups, but review and evaluation is included into control process group).

References

6 Jason Charvat *Project Management Methodologies : Selecting, Implementing, and Supporting Methodologies and Processes for Projects*

7 PMI Institute *A Guide to the Project Management Body of Knowledge*

8 Lecky-Thompson *Corporate Software Project Management*

9 International standard ISO/IEC 12207:1995(E)

10 C. Rolland *A comprehensive view of software engineering*

11 K. Van Stolen. B. Hodes *Characterizing IS development project*

12 Object Management Group *Software Process Engineering Metamodel*

13 WWW, Max Wideman, *Software development and linearity,* http://www.maxwideman.com/papers/linearity/pmbok.htm

14 Pekka Abrahamsson, Outi Salo, Jussi Raikannen, Juhani Warsta, *Agile software development methods*

15 David J. Schultz *A comparison of five approaches to software development*

16 W. W. Royce *Software management renaissance*

17 Barry W. Boehm *A Spiral model of Software Development and Enhancement*

18 W. W. Royce *Managing the development of large software systems*

19 R. M. Wideman *Comparing PRINCE 2 with PMBOK*

20 Young Hoon Kwak, C. William Ibbs *Project Management Process Maturity Model*