Planning of Project Scope and Work

In the project context planning of project work means planning of activities, time usage, required personnel and equipment, technology and costs to implement scope agreed with product owner. In the context of information system development means planning of project work planning of information system change processes and required inputs (resources) to accomplish these processes. Pictorially expressed on the next figure:

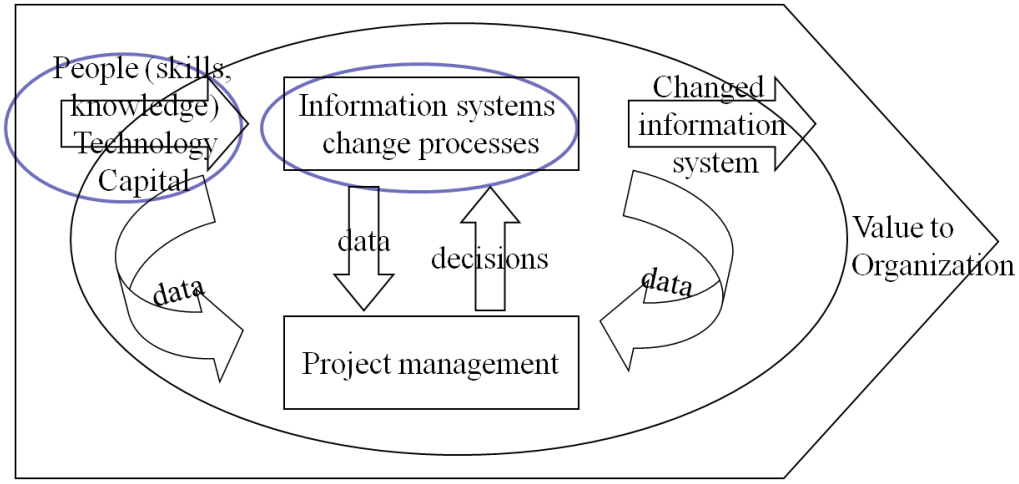


Figure 1. Planning Project Work in the Context of Information System Development

Project scope and time usage planning are the subjects of this lecture. Also I finish with describing scope planning remaining processes – scope defining and creating work breakdown structure (WBS).

Project Scope

Project scope deals with how one or another expected deliverable will be created and delivered to customer. It deals with deliverables and project management documentation describing all the work we must accomplish to achieve project goals. Depending on customer technical experience, documentation can be more or less complex or formal, for example:

- data model where are shown main data entities with relations or logical data model where data are normalized with 3. normal form
- application specification which consists of manually written sentences on paper or of to the very last elaborated diagram showing program structure

Different choices demand different amount of work

Additionally – in context of system development, project scope is dictated from system development methodology determining, what kind of documentation is needed, in which sequence and content. With project scope we must determine, what kind of methodology customer uses and how strictly it will be followed. If client considers that following methodology is costly, even so it is necessary to follow it in appropriate level when to consider project success.

One example of quantity of rules to be followed by certain methodology showing on the next figure:

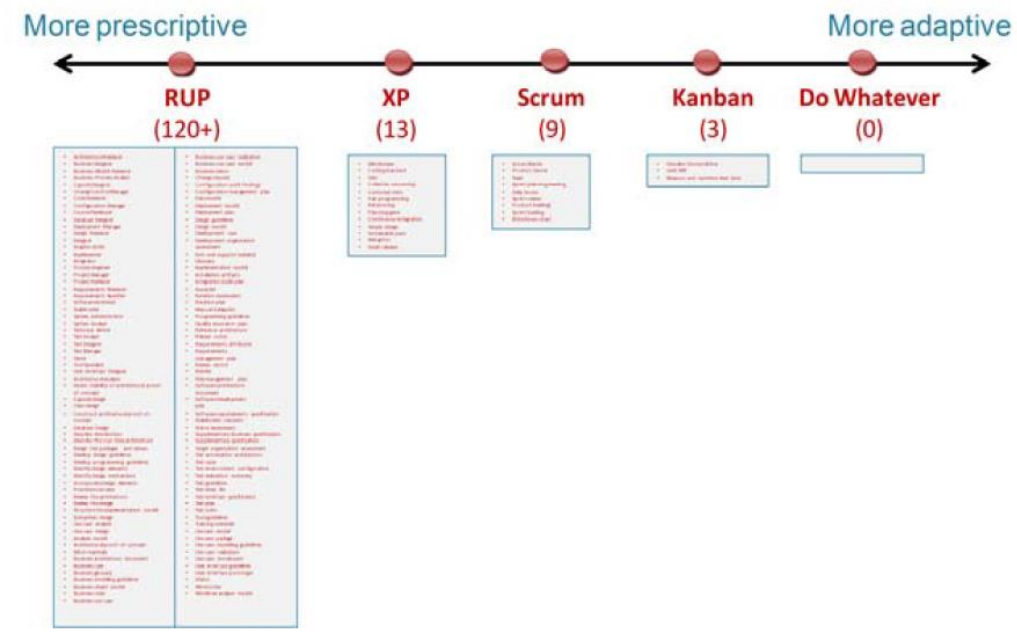


Figure 2. Methodology Rules Quantity Difference

In context of system development project scope by RUP includes:

- customer products (things what customer needs for useful system, in addition to program code also user manuals, help files, installation scripts and guides)
- process artifacts (things what development team during development creates, for example use case specifications, design documentation etc.)
- internal deliverables (things having value to development organization – development results consisting intellectual value, for example requirements documentation templates, test plans etc.)
- services (work what is expected in addition to delivered product or development of another results, for example training, consultation, installation, onsite support and adaptation to customer)

Criteria for Choosing System Development Methodology

- Nature and scope of system under development

- Project criticality
- Budget
- Team Size
- Used Technology
- Used Tools and Techniques
- Work culture in organization

Nature of the Software Developments

Steve McConnel in his book "Rapid Development" differentiates basic lifecycle approaches to 3 groups:

- Demos and “proof-of-concept” prototyping - needs some conceptual modeling to guide the coding, but not detailed requirements, design, or testing
- New technology application delivery - requires an incremental approach, iterating between the software development phases of architecture, detailed design, coding, and testing
- Stable application maintenance - needs a sequential waterfall approach repeating a well-understood process for a well-understood product or application

Adaptive (“agile”) versus Predictive (“plan-driven”) Approach

- Adaptive/ agile
  - Low criticality
  - Senior developers
  - Requirements change often
  - Small number of developers
  - Culture that thrives on chaos
- Predictive/ plan-driven
  - High criticality
  - Junior developers
  - Requirements do not change often
  - Large number of developers
  - Culture that demands order

Boehm and Turner Radar Chart

- Personnel - measures team skills
- Dynamism - likelihood of changes
- Culture - temperament of the organization – thriving on chaos vs. order
- Team Size
- Criticality - system failure results in loss of …

Pictorially expressing:

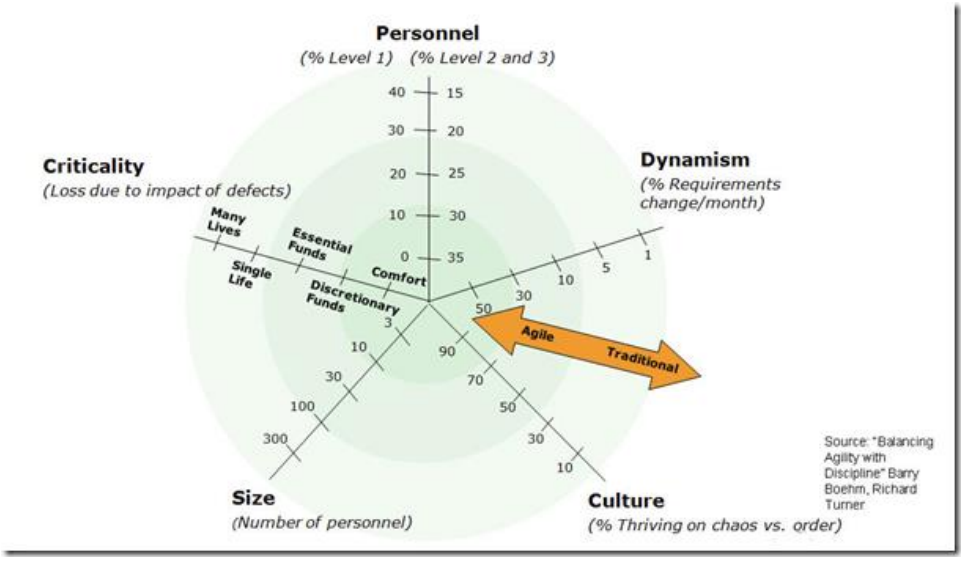


Figure 3. Boehm and Turner Radar Chart

Instead of Summary of this Topic

Henrik Kniberg and Mattias Skarin in their book “From Kanban and Scrum. Making the Most of Both” wrote following:

- All system methodologies are process tools
  - Tools – anything used as a means of accomplishing a task or purpose
  - Process - how you work
- They help you work more effectively by, to a certain extent, telling you what to do
- Using the right tools will help you succeed, but will not guarantee success
- It's easy to confuse project success/failure with tool success/failure
  - A project may succeed because of a great tool
  - A project may succeed despite a lousy tool
  - A project may fail because of a lousy tool
  - A project may fail despite a great tool
- We can compare process tools by looking at how many rules they provide where prescriptive means “more rules to follow” and adaptive means “fewer rules to follow”
  - 100% prescriptive means you don’t get to use your brain, there is a rule for everything
  - 100% adaptive means Do Whatever, there are no rules or constraints
- No tool is complete, no tool is perfect
- Compare for understanding, not for judgment
- Don’t limit yourself to one tool!
- Mix and match the tools as you need!

Defining Scope

Defining scope is creating project scope statement. It is a process of developing a detailed description of the project and product. Inputs for this process are project charter, requirements documentation and organization process assets (templates for example). The output of this process is project scope statement. The preparation of a detailed project scope statement is critical to project success and builds upon the major deliverables, assumptions, and constraints that are documented during project initiation. During planning, the project scope is defined and described with greater specificity as more information about the project is known. Existing risks, assumptions, and constraints are analyzed for completeness; additional risks, assumptions, and constraints are added as necessary. If something isn't described within the detailed project scope statement then the work should not be done or the scope statement needs revised to include the work

Project Scope Statement

Describes, in detail, the project's deliverables and the work required to create those deliverables; **provides** a common understanding of the project scope among project stakeholders; **may contain** explicit scope exclusions that can assist in managing stakeholder expectations; **enables** the project team to perform more detailed planning, **guides** the project team's work during execution, **provides** the baseline for evaluating whether requests for changes or additional work are contained within or outside the project's boundaries.

The degree and level of detail to which the project scope statement defines the work that will be performed and the work that is excluded can determine how well the project management team can control the overall project scope

Project scope statement should consist at least following components (they may be written in separate documents:

- Product scope: The characteristics of the product, service, or result for which the project was undertaken. In projects that are part of a larger program, the project itself may only be creating components of the product, but the product scope or product description is still necessary so that everyone knows what the overall objective is.
- Project objectives: Objectives are the success metric for the project. Specifically, what will it take for the project to be considered successful? This includes the business, cost, schedule, technical, and quality objectives, and other specific targets should be included where applicable.
- Project requirement: The capabilities that the product, service, or result must possess and meet. Requirements are the translated expectations and needs of the stakeholders into prioritized, descriptive requirements and work items.

- Project exclusions: Nearly just as import as what IS in the project, the scope should include items that are excluded from the project. Doing this helps eliminate any confusion within the stakeholders or project team.
- Project deliverables: The core product, service, or result should be fully described, as well as any ancillary deliverables. Any needed project artifacts, those documents not directly related to the deliverables, such as management, technical, or status reports, should also be described.
- Product acceptance criteria: The process and criteria for product acceptance should be defined. This includes customer-specific requirements and any testing or other threshold limits.
- Project constraints: Any limiting factors the project must work within, such as deadlines, budget, staffing, facilities, equipment, materials, or contractual restraints, should be described.
- Project assumptions: Every project has assumptions, and these should be described because assumptions are risk factors.
- Risks: Risks should be identified at least at a high level. The risk register is where all risks are logged, but having major risks explained in the scope statement helps make everyone aware and on the lookout for them.
- Milestones: Any important dates, including deliverable- or artifact-oriented dates should be included in the project scope statement.
- Approval requirements: Any specific approval requirements for items such as deliverables, documents, and work should be described.

Creating Work Breakdown Structure

Deliverables-oriented, graphical, hierarchical representation of the work required to fulfill the project scope statement. Purposes:

- it subdivides the work into manageable components that can be scheduled, estimated, and assigned
- through the process of creating and updating the WBS, it helps to identify needed work that might otherwise not have not been discovered until later
- it can be used as a visual communication tool for the customer, stakeholders, and project team
- the WBS is an input to activity definition, cost estimating, cost budgeting, resource planning, and risk management planning

Inputs for creating WBS are project scope statement; requirements documentation and organizational process assets (templates). Outputs are WBS, WBS dictionary, scope

baseline and project documents updates. Tool and technique for that process is decomposition.

WBS Design Principles

- Project scope is divided into manageable components in terms of size, duration, and responsibility (e.g., systems, subsystems, components, tasks, subtasks, and work packages) which include all steps necessary to achieve the objective – result is hierarchy or tree
- WBS includes 100% of the work defined by the project scope and captures ALL deliverables – internal, external, interim – in terms of the work to be completed, including project management
- The sum of the work at the "child" level must equal 100% of the work represented by the "parent"
- WBS should not include any work that falls outside the actual scope of the project, that is, it cannot include more than 100% of the work…
- It is important that there is no overlap in scope definition between two elements of a Work Breakdown Structure

Examples of creating WBS structure:

- Using phases of the project life cycle as the first level of decomposition, with the product and project deliverables inserted at the second level;
- Using major deliverables as the first level of decomposition;
- Using subprojects which may be developed by organizations outside the project team, such as contracted work - the seller then develops the supporting contract work breakdown structure as part of the contracted work

Example of WBS pictorially:

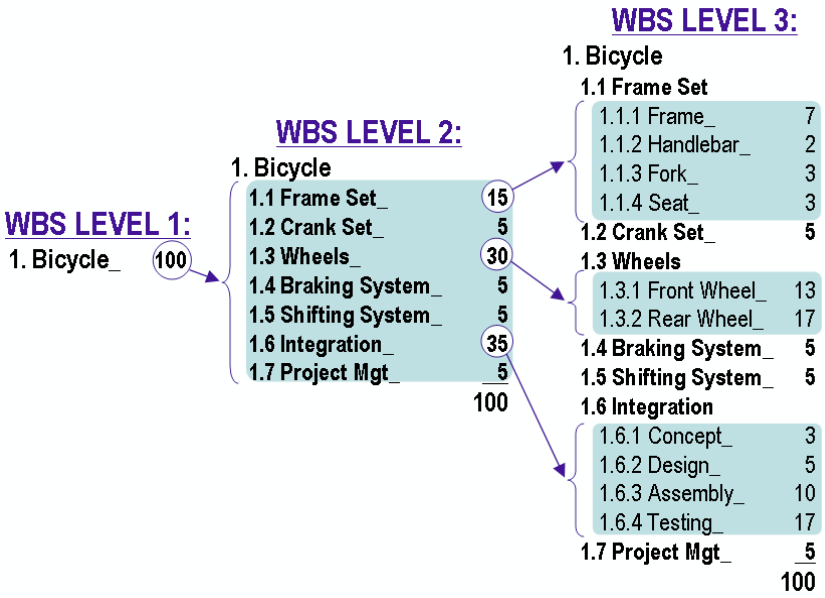


Figure 4. Example of Bicycle Work Breakdown Structure

WBS Dictionary

Provides more detailed descriptions of the components in the WBS, including work packages and control accounts:

- Code of account identifier
- Description of work
- Responsible organization
- List of schedule milestones
- Associated schedule activities
- Resources required
- Cost estimates
- Quality requirements
- Acceptance criteria
- Technical references
- Contract information

Scope Baseline

A component of the project management plan includes project scope statement; WBS and WBS dictionary.

Project Time (Usage) Planning in PMBOK

Putting work on time axis including

1. defining activities
2. sequencing activities
3. estimating activity resources
4. estimating activity durations
5. developing schedule

Pictorially expressing:

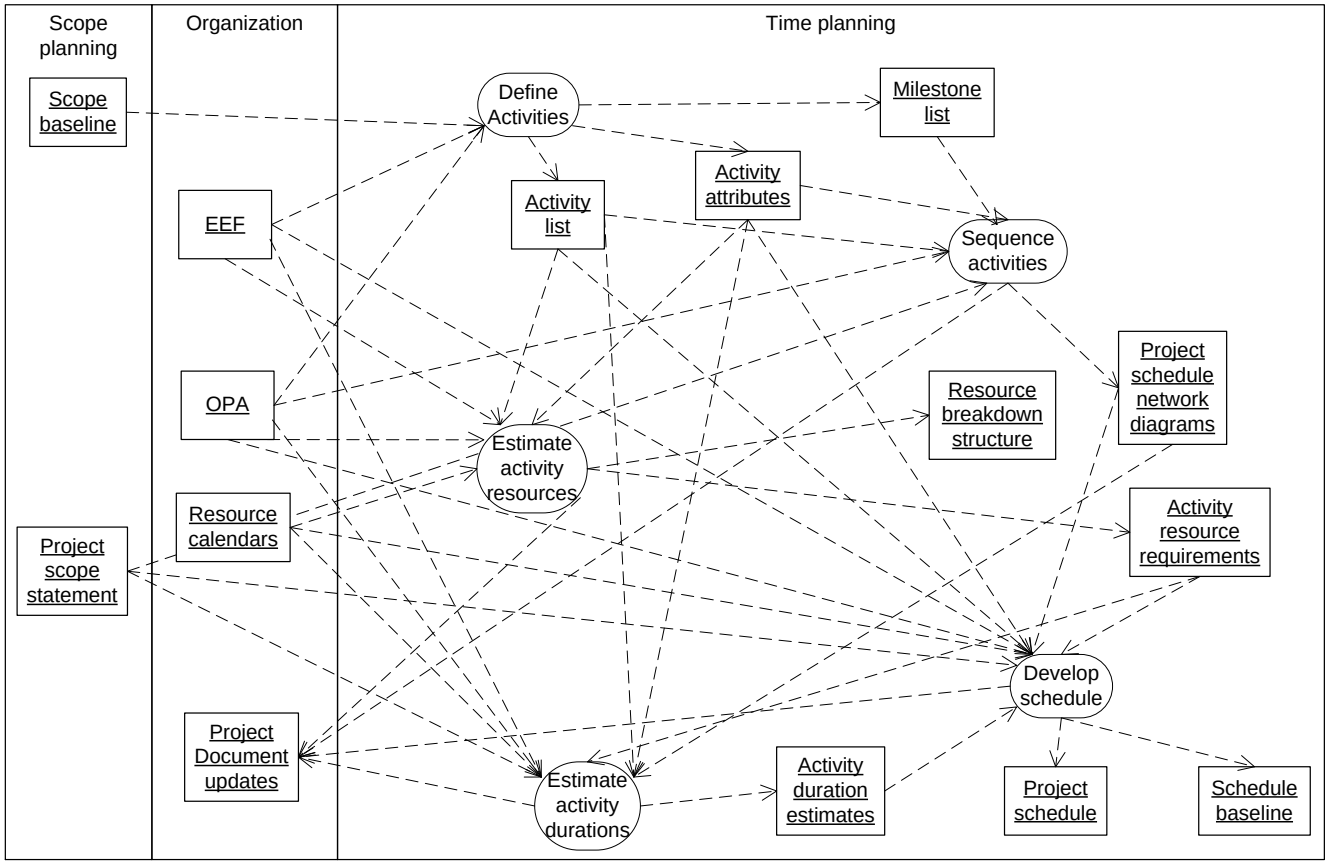


Figure 5. Project Time Planning Activities

The basis for time planning:

- project scope –
  - product conceptual design +
  - requirements for achieving project objectives
- work / deliverables breakdown structure
- arrangement of work processes
  - methodology or development strategy

There can be different development strategies to choose amongst:

- Incremental: Slow but steady approach (without attempting a leap) in which an already conceived end result is aimed for.
- Evolutionary: Slow but steady approach (without attempting a leap) in which there is no pre-conceived end result but each successive design or product is a refinement of the previous one.
- Grand design: Total transformation through a right-the-first-time approach.

Activities definition and sequencing processes and their mutual relationships are presented on the next figure:

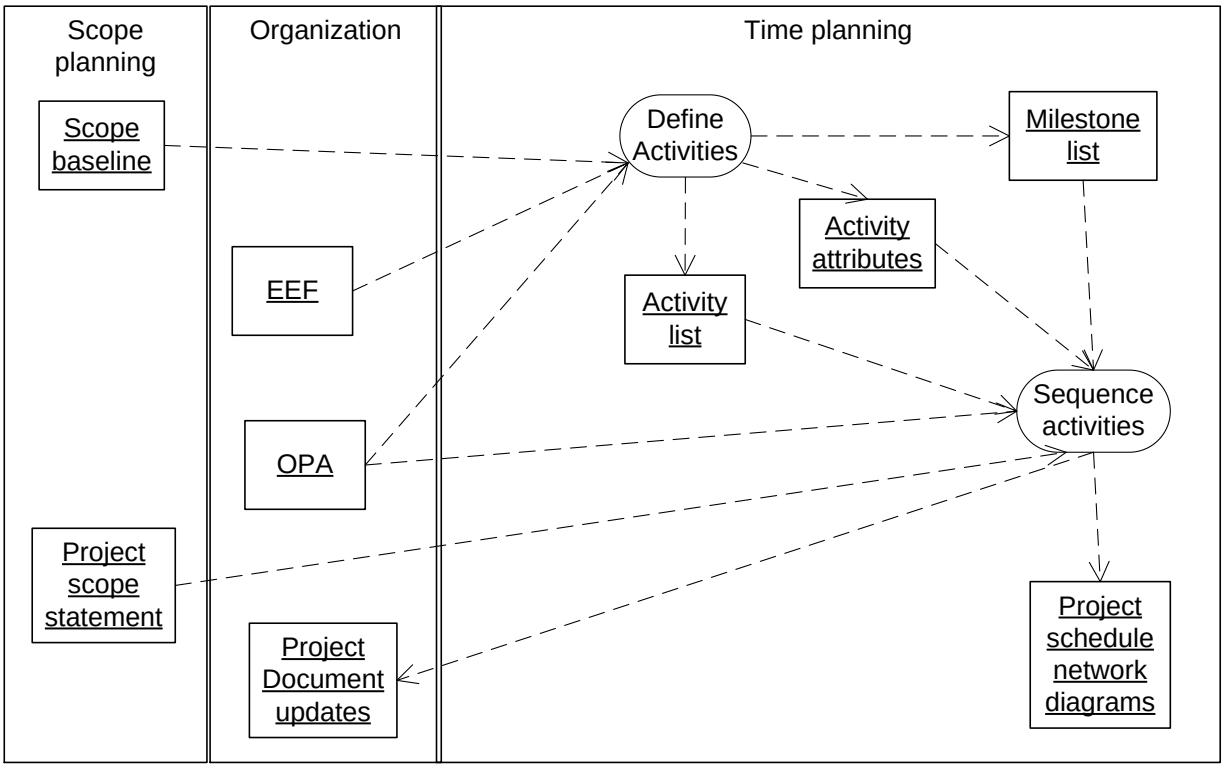


Figure 6. Activities Defining and Sequencing Processes and Their Inputs and Outputs

Step 1 – defining activities

Decomposing the WBS work packages into activities where work packages are evaluated for how they can be broken down into manageable activities.

The output of that process is the complete list of project activities - activity list - each activity should have a unique identifier that correlates it to the WBS work package

An adequate level of activity decomposition is generally reached when the activities:

- are assignable to one person
- can have a level of effort determined for them
- can have their resource needs estimated
- can have their expected costs reasonably established

- can have their progress determined and tracked

Every activity must be fully described containing:

- explanations of any special dependencies or relationships that exist
- assumptions that were made
- person responsible for the activity
- special equipment or materials needed
- information specific to the activity that will vary depending upon the project's application area and the type of activity

Activity types categorize how measurable the activity is and how it's related to the project's objectives:

- discrete effort - a discrete effort activity is one whose work directly relates to a work package or deliverable on the work breakdown structure. These types of activities need to be measurable since they tie directly to the project's core objectives.
- level of effort (or LOE) - is usually an activity performed by a supporting role that is difficult to measure, but is still related to the project's core objectives.
- apportioned effort (or AE) - activity is one which is usually related to project management –it's necessary for the efficient functioning of the project, but it isn't directly related to the project's final product, service, or result.

Another important output of activity definition is the milestone list which provides all significant events and dates on the project. Time periods between milestones are defined as phases or iterations

Defining Milestones

Milestones provide a very effective method for communicating the schedule progress to stakeholders. They are description of the states of the project at a certain point of time. They can be points at which major deliverables are completed, phases are reached, or other important dates. The last milestone in project is the point where project stakeholders formally decide to accept the ownership of project results

Milestone describes what the project should achieve, not how. It should be neutrally formulated with regard to the solution - we must have freedom to choose the activities to reach the desired state, for example: "members have specified knowledge in a stated area" or "members have completed course X"

There are major milestones – important events to customer and minor milestones – important events to project team(s)

Examples of software development milestones are presented in the following table:

Table 1. Example of Milestones in Software Development

Milestone	Criteria
Feature freeze	The date when all required features are known and the detailed design has uncovered no more. No more features are inserted into the product.
Code freeze	Implementation of the design has stopped. Some testing of the features has occurred.
System test freeze	Integration testing is complete. Code freeze for system test to start.
Beta ship	The date the Beta software ships to Beta customers
Product ship	The date the product ships to the general customer base

Defining Phases

The time between two major project milestones, during which a well-defined set of objectives is met and artifacts are completed – group of activities. They provide project stakeholders with the opportunity to make significant decisions about committing resources to the project, one step at a time.

Each phase must be concluded with a formal milestone decision - the point at which stakeholders must decide to commit resources to achieve the objectives of the next phase (not the entire project!)

Phases in turn are refined in smaller time periods (iterations) having same features than phases.

Step 2 – sequencing activities

Sequencing activities involves determining the dependencies and relationships between activities and applying leads and lags. The result is a project network - logical structure of successive project activities at the end of what must be achieved certain project objectives/results.

Activity sequences are shown through project schedule network diagrams which schematics are showing the order and relationships of project activities. Dependences between activities may be

- hard logic – we can perform an activity when preceding activity or activities are finished
- related with resources availability – performing an activity is possible when needed resources are available

Leads and lags are artificial effects on the timing of activities and are used in conjunction with activity relationships and dependencies. Leads and lags and the

reasoning behind their use should be well documented because it may not be obvious to the casual observer why they were established

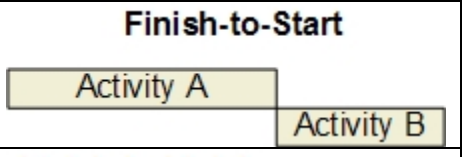
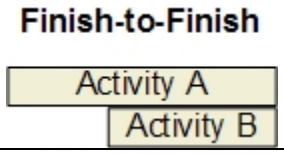
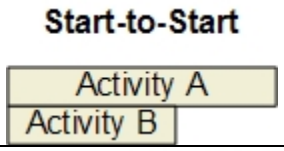
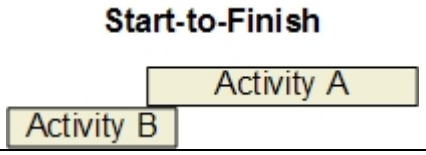
Lags are delays or waiting time between activities - positive time added to an activity's duration

Leads speed up activities without changing the relationships between the activities - “negative” time because they allow activities to occur in parallel that would normally be done sequentially

Task Link Types

They are presented in the following table:

Table 2. Task Link Types

Name	Semantics	Pictorially
Finish – to – start (FS)	A must finish before B starts	
Start – to – start (SS) –	A must start before B can start	
Finish – to – finish (FF)	A must finish before B can finish	
Start – to – finish (SF)	A must start before B can finish	

(Time (Usage)) Planning in RUP

Main source for following material is The Rational Edge 2003 article „Project Planning Best Practices“. One best practice for project planning is multi-level planning, or developing plans at different levels of detail. These levels are necessary because not all stakeholders are interested in the same information. The combination of high-level (top-down) and detailed (bottom-up) planning is key to iteratively refining the Project Plan.

You can develop a Project Plan by performing a set of basic activities:

1. Define project objectives
2. Define phases
3. Partition project effort and schedule across phases
4. Prioritize risks

5. Define milestones
6. Define iterations
7. Partition project effort and schedule across iterations

In general, you must perform these activities repeatedly, and not necessarily in the sequence shown here. In current lecture are described activities 2, 3 and 6. In the following lecture I handle activities 3 and 7.

Phases Definition

In this activity you describe the project's lifecycle. Phases provide project stakeholders with the opportunity to make significant decisions about committing resources to the project, one step at a time. Each phase must be concluded with a formal milestone decision -- the point at which stakeholders must decide to commit resources to achieve the objectives of the next phase (not the entire project!). The last project milestone is the point at which stakeholders decide to formally accept ownership of the project's results. RUP phases and their objectives are presented in the next table:

Table 1. RUP Phases and Their Objectives

Phase	Objectives
Inception	
	Gain stakeholder agreement on scope and boundary conditions
	Economically motivate the project.
	Ensure mutual understanding of critical functionality.
	Produce overall estimates of cost and schedule.
	Gain insight into risks that may threaten project success.
Elaboration	
	Demonstrate that the system architecture will satisfy the significant constraints imposed on the product.
	Achieve stability of the product vision.
	Produce a realistic plan for executing the remainder of the project.
	Eliminate any significant risks that may threaten the success of the project.
Construction	
	Build and test the system in increments.
	Ensure increasing understanding and acceptance of the system by stakeholders.
	Minimize costs and schedule.
Transition	Achieve user self-sufficiency.
	Achieve acceptable operational quality.
	Transfer ownership of the system to support and maintenance.
	Evaluate and conclude project.

Iteration Definition

Iteration is defined as a time-boxed period during which one or more teams perform a set of related activities to produce a coherent and (partially) complete product. The rule of thumb by defining iterations in RUP phases: in inception and transition phase 1

iteration, in elaboration and construction phase 2 iterations, or divide Elaboration and Construction into 4-week iterations and Inception and Transition into 6 week.

Iteration Planning

Near the end of each iteration, you should create a plan for the next iteration. The result of iteration planning is a fine-grained plan for a specific iteration. This plan contains the details omitted in the overall Project Plan: artifacts, activities, names, dates, and required effort. Iteration planning consists of the following basic activities:

1. Refine iteration objectives
2. Develop product breakdown structure
3. Define evaluation criteria
4. Define iteration milestones
5. Partition effort across disciplines
6. Develop work breakdown structure
7. Staff team and assign work
8. Consolidate planning

With following I give overview about activities 1, 3 and 6.

Defining Iteration Objectives

Every iteration should have specific objectives - they should eliminate risks. Example of iteration objectives is shown in the following table:

Table 3. RUP Iterations Objectives Example

Iteration	Objectives
Inception-1	Achieve stakeholders' concurrence on **scope**
	Refine **estimates** based on actual experiences during Inception.
	Identify and prioritize **risks**.
	Construct a detailed **plan** for the remainder of the project.
Elaboration-1	Develop a **skeleton solution** which covers the complex, volatile, and risky parts of the system.
	Eliminate highest priority technical **risks**.
	Ensure project team proficiency with **process and tools**.
	Ensure stability of the development and test **environments**.
Elaboration-2	Develop a **skeleton solution** with broad functional coverage and depth in only the risky parts.
	Eliminate all known nontrivial **risks**.
	Construct a detailed **plan** for the remainder of the project.
Construction-1	Develop **80 percent of the solution** as rapidly as possible with reasonable quality.
	Minimize **costs** by optimizing schedule and resources.
Construction-2	Develop **100 percent of the solution**.
	Achieve **acceptable quality** of all developed portions of the solution.
	Prepare **deployment** to the end-user and support organization
	Minimize **costs** by optimizing schedule and resources.
Transition-1	Improve the **quality** of the solution.
	Ensure end-user organization **ability to use** the solution.
	Ensure support organization **ability to support** the solution.
	Close out the project.

Defining Iteration Evaluation Criteria

In order to define the desired end state of the iteration, you must formulate iteration evaluation criteria. From a quality perspective, the purpose of evaluation criteria is to provide a means of measuring the success of an iteration. Within the context of planning, defining evaluation criteria is a form of the iterative project management best practice: Begin with the end in mind.The set of evaluation criteria should clearly define the end -- or success criteria -- of the iteration. Group the criteria by the appropriate discipline, as shown below (example).

- Requirements
  - Is the Vision stable?
  - Did stakeholders actively participate in reviewing the Vision?
  - Is the requirement set complete?
  - Are all inconsistencies resolved?
  - Are traceability relations and requirements attributes up to date?
- Analysis and design
  - Is the software architecture stable?
  - Does the design satisfy the quality ranges defined in the design guidelines?
  - Has the design of the targeted use cases been documented and reviewed?
  - l Have the design guidelines been adhered to?

Developing Iteration Work Breakdown Structure

WBS is developed in 3 levels:

1. level – iteration
   a. 2. level – discipline
      i. 3. level – activity

Iteration WBS example is illustrated on the next figure:

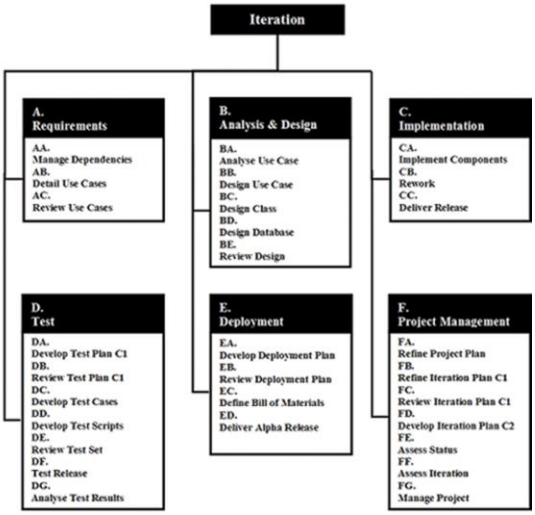


Figure 7. Sample Iteration WBS

Used Literature

- Definition of Development Strategy, http://www.businessdictionary.com/definition/development-strategy.html
- IEEE Guide--Adoption of the Project Management Institute (PMI(R)) Standard A Guide to the Project Management Body of Knowledge (PMBOK(R) Guide)--Fourth Edition, http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6086683
- Scope Planning Templates, http://www.projectmanagementdocs.com/project-planning.html
- Jolyon E. Hallows: Information Systems Project Management, 2. Ed., http://atoz.ebsco.com/Titles/SearchResults/1201?SearchType=Contains&Find=Information+Systems+Project+Management&GetResourcesBy=QuickSearch&resourceTypeName=booksOnly&resourceType=8%2C9%2C14&radioButtonChanged=
- Simon Harris, The Breakdown Structure: 90% of how to manage projects safely, http://www.asapm.org/asapmag/articles/TheBreakdownStructurePt1.pdf ja http://www.asapm.org/asapmag/articles/TheBreakdownStructurePt2.pdf
- WBS Examples
  - http://en.wikipedia.org/wiki/Work_breakdown_structure
  - http://www.pmhut.com/project-management-process-phase-2-planning-develop-project-schedule
- WBS Review, http://cvisn.fmcsa.dot.gov/downdocs/cvisndocs/plan99/2%20Day%200%20Sessions%20for%20update/P2%20Intro%20to%20WBS%20Development_R26.ppt
- RUP, http://sce.uhcl.edu/helm/rationalunifiedprocess/index.htm
- P. Fitsilis, Comparing PMBOK and Agile Project Management Software Development Processes, http://www.springerlink.com/content/j52npr0157v26386/
- William J. Brown, Hays W. McCormick III, Scott W. Thomas: Antipatterns in Project Management, 2000
- Alistair Cockburn, Methodology per project, http://alistair.cockburn.us/Methodology+per+project
- Mike Griffiths, **“**Agile Suitability Filters**”,** http://leadinganswers.typepad.com/leading_answers/files/agile_suitability_filters.pdf
- Henrik Kniberg and Mattias Skarin**, „**Kanban and Scrum – making the most of both“, 2009, http://www.infoq.com/minibooks/kanban-scrum-minibook

- John Filicetti. Project Management Process - Phase 2 - Planning - Develop Project Schedule, http://www.pmhut.com/project-management-process-phase-2-planning-develop-project-schedule
- Plan and Schedule Development, http://www.nehimss.org/070518_Project_Management/Chance_Reichel_Templ_TaskID_WBS_05_12_00.doc
- Per Kroll, Planning and estimating a RUP project using IBM Rational SUMMIT Ascendant, http://www.ibm.com/developerworks/rational/library/4772.html
- Eric Lopes Cardozo, DJ de Villiers. Project planning best practices, The Rational Edge, August 2003, http://www.ibm.com/developerworks/rational/library/content/RationalEdge/aug03/m_projectplanning_dd_ec.pdf
- Johanna Rothman, Iterative Software Project Planning and Tracking, http://www.jrothman.com/Papers/7ICSQ97.html