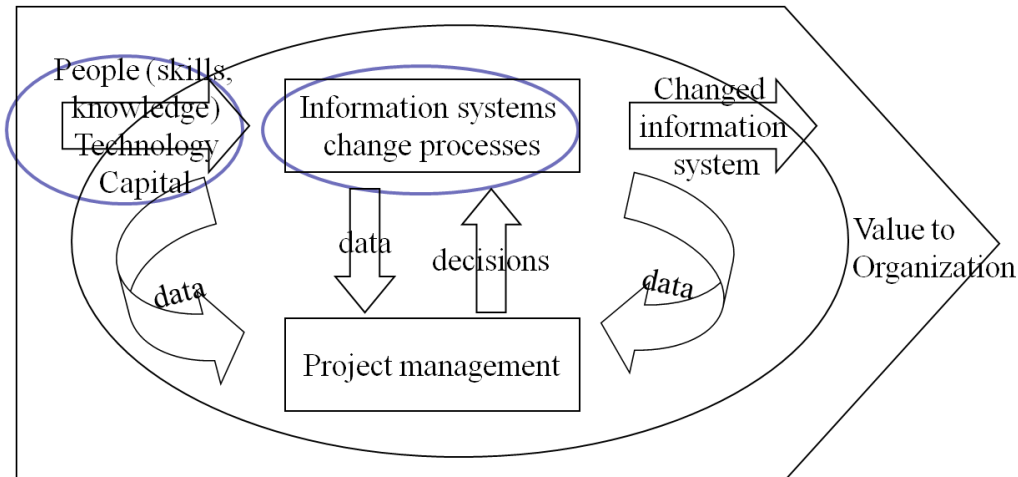


## Planning Project Work

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 to accomplish these processes. Pictorially expressed on the next figure:



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

For success of project management it is very important that scope agreed with product owner is consistent with usable time and costs. This principle is "project triangle" indicating that usable time and costs must be in correspondence with scope to implement. It is project manager responsibility to create and hold this situation. Project manager must plan work processes and required resources to implement project scope. Time and cost planning is the subject of this lecture.

## 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:

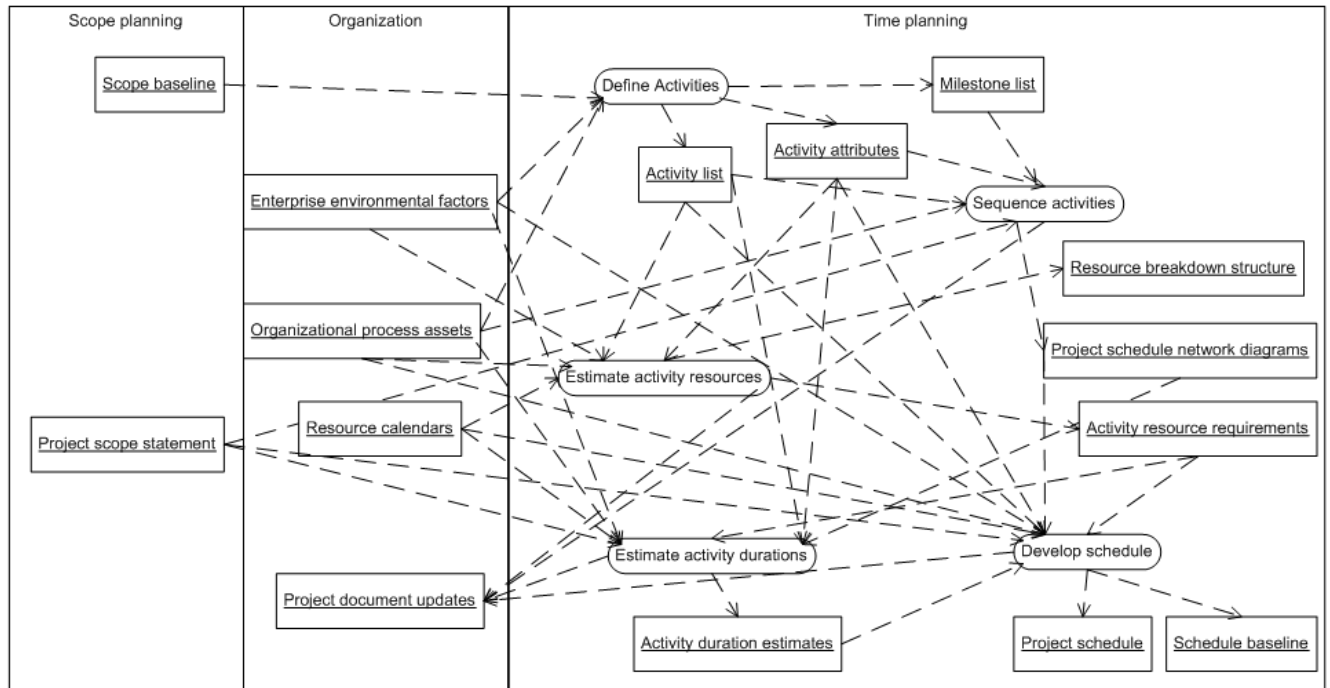


Figure 2. Project Time Planning Activities

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.

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

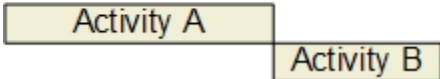
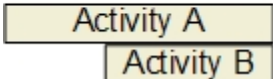
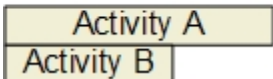
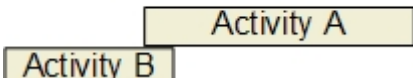
Lags are delays or waiting time between activities - positive time added to an activity's duration

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	<b>Finish-to-Start</b> 
Start – to – start (SS) –	A must start before B can start	<b>Finish-to-Finish</b> 
Finish – to – finish (FF)	A must finish before B can finish	<b>Start-to-Start</b> 
Start – to – finish (SF)	A must start before B can finish	<b>Start-to-Finish</b> 

Step 3 – estimating activity resources

Identifying the quantities of resources needed for the project’s activities. Resources are anything having a monetary cost and include materials, equipment, licenses, fees, and personnel needed for the project’s activities.

The quantity designations will depend upon the type of resource:

- resource quantities for personnel may be expressed in hours, work days, or work weeks
- the quantity designations for materials to be consumed will be expressed in whatever unit is applicable (cases, boxes, pounds, tons, kilograms, and so on)
- quantity designations for equipment or facilities would likely be expressed as a unit of time (hours, weeks, or months)

### Estimating Methods

- Top-down estimating form of expert judgment through which an item is looked at broadly and a generalized estimate created
- Bottom-up estimating provides estimates for each component activity, and then aggregates these into an overall estimate

### Estimating Accuracy

- Rough order of magnitude estimate - these are usually top-down estimates made by expert judgment. The variance range for this type of estimate is expected to be -25% to +75% of the final actual figure
- Budget estimate - these have less variance than rough order of magnitude, but they are still broad estimates. The variance range for this type of estimate is expected to be -10% to 25% of the final actual figure
- Definitive estimate - this type is the most accurate estimate. The variance range for this type of estimate is expected to be -5% to 10%

### Estimating Project Effort

The labor required to deliver a product of a given size. Effort is measured in person-hours, person-days, or person-months. For example: for strategic analysis 500 hours or 60 work days or 3 months

### Effort Estimation Methods

Calculating size - sizing by analogy or sizing by analysis where

- sizing by analogy is based on experiences from previous projects.
- sizing by analysis - Function Point Counting, Feature Points, Predictive Object Points, Use Case-s, Story Points etc

Algorithmic software cost estimation model COnstructive COst MOdel (COCOMO) - program size is expressed in estimated thousands of lines of code (KLOC)

### Step 4 – estimating activity duration

How long the activity will take expressed in a work period (day, hour). Usually number of days where 1 or more resources actually at the activity work

The bases for duration estimation are size of effort, number of resources and work hours of every resource per work day

### Estimation Methods

- Using calculation method: duration = effort size/ resource quantity / resources work hours number per work day, for example: 10 hours effort / 2 people / 2 hours per work day = 2,5
- Using PERT (Program Evaluation and Review Technique) method (calculating weighted average duration) - WAE =  $(OE+4*MLE+PE)/6$

### Step 5 – schedule development

Develop Schedule is the process of analyzing activity sequences, durations, resource requirements, and schedule constraints to create the acceptable (realistic) project schedule. An approved project schedule can serve as a baseline to track progress. Schedule development is an iterative process because project manager must periodically revise this schedule according real life corrections.

Inputs for the schedule development are outputs from previous steps (activity definitions, sequencing etc); resource calendars (personnel and equipment availability times for the project; project scope statement; project calendar (available time to completion of project work).

Tools and techniques for schedule development are for example:

- critical path method
- critical chain method
- resource leveling
- schedule compressing

## Planning in RUP

Planning and estimating in RUP project are divided into the following three main activities:

- Perform initial project sizing - understanding a project's size is part of the start-up. This allows understanding the project in terms of work effort and provides a basis for high-level resource planning by skill type. Project manager can periodically revisit the initial project sizing to refine it as he/she learn more about project characteristics
- Produce a phase plan - in the first iteration a high-level phase plan is produced that provides initial cost and schedule "scale" information for a project go/no-go decision or possible re-scoping. Later, project manager can use this high-level

plan to check and refine the overall scale of the project using the actual results of each iteration as input

- Produce iteration plans - for each iteration an iteration plan is produced and used to guide work and track progress. The iteration plan specifies what deliverables it should be produced, who creates them, and how to assess the results, primarily through executable testing.

Pictorially presented on the next figure:

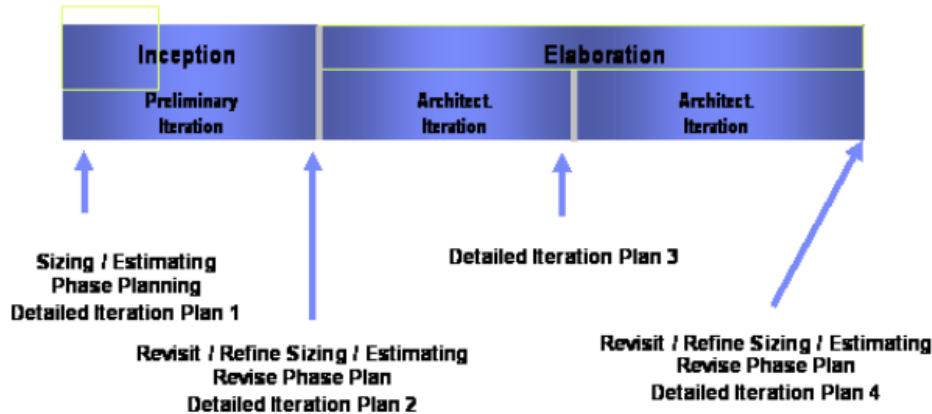


Figure 3. Planning Activities in RUP

Iteration Definition

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.

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



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

Partition Project Effort and Schedule Across Phases

Is presented in the next table:

**Table 4. Partition of Effort and Schedule in RUP Phases**

Phase	Start date	End date	Schedule	Effort
Inception	2003-01-06	2003-01-31	4 weeks (15%)	600 hours (8%)
Elaboration	2003-02-03	2003-03-14	6 weeks (21%)	1400 hours (17%)
Construction	2003-03-17	2003-06-06	12 weeks (43%)	4800 hours (60%)
Transition	2003-06-09	2003-07-18	6 weeks (21%)	1200 hours (15%)
<b>Total</b>			<b>28 weeks</b>	<b>8000 hours</b>

Partition Phase Effort and Schedule Across Iterations

In order to get schedule and effort of one iteration. In partitioning schedule, it is usually easiest to divide the phase into iterations of equal length or first iterations longer. Same way with partitioning effort. Iterations of equal length within a phase usually require equal effort. Allocate more effort to longer phases and less to shorter ones, so that the total effort per week remains relatively constant within a phase. Pictorially expressing:

Table 5. Partition Phase Effort and Schedule Across Iterations

Phase	Iteration	Schedule	Effort
Inception	I1	4 week	600 hours
Elaboration	E1	3 weeks	700 hours
	E2	3 weeks	700 hours
Construction	C1	6 weeks	2400 hours
	C2	6 weeks	2400 hours
Transition	T1	6 weeks	1200 hours

**Agile Planning and Estimation**

- Product level – 1-2 times in the year – attention on product creation in time
- Release level – 3-4 times in the year – attention on compromises concerning features and delivery dates
- Sprint level – in every iteration – attention on features under development in the frames of sprint
- Every day level – every day – attention on everyday commitment in completing features.

Pictorially expressed on the next figure:

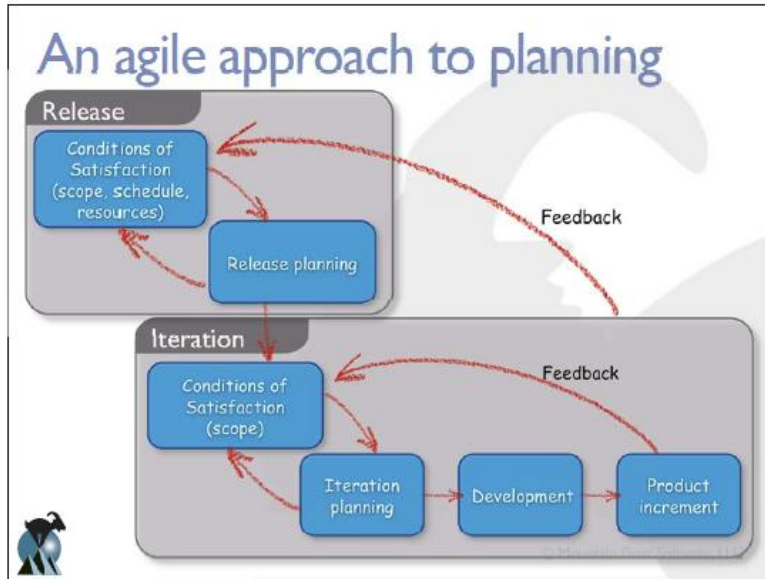
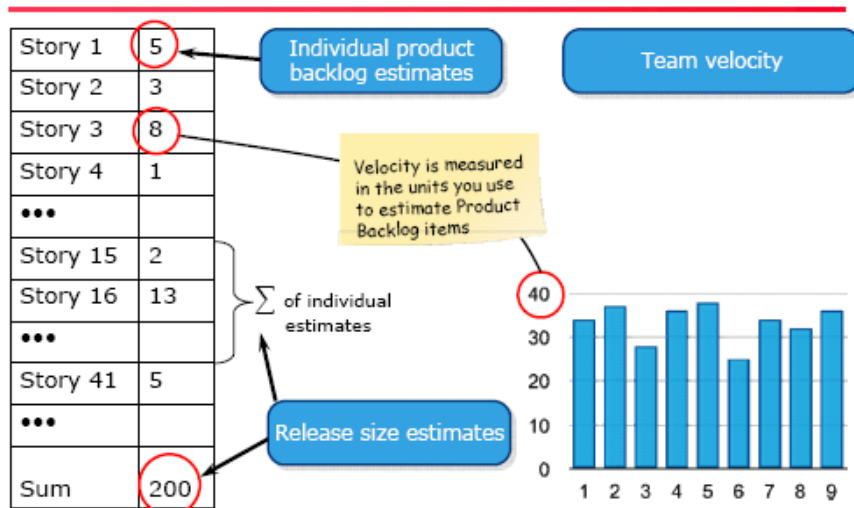


Figure 1. Agile Approach to Planning

Main Elements in Agile Planning

- User Story and individual product backlog estimates (Story Points)
- Release size estimates
- Velocity – is measured in the units you use to estimate Product Backlog items
- Team velocity
- Planning Poker

Pictorially:



Copyright (c) 2006-2007, Innolution, LLC and Mountain Goat Software, Inc. All Rights Reserved.

Joonis 2. Elements in Agile Planning

Requirements Management in Scrum

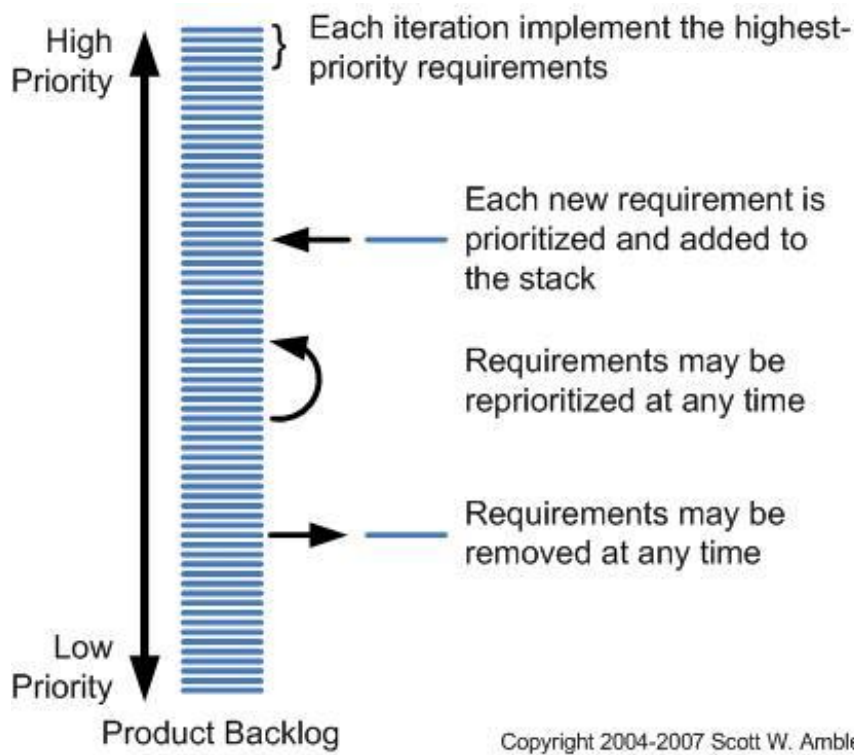


Figure 3. Requirements Management in Scrum

Disciplined Agile Work Management Process

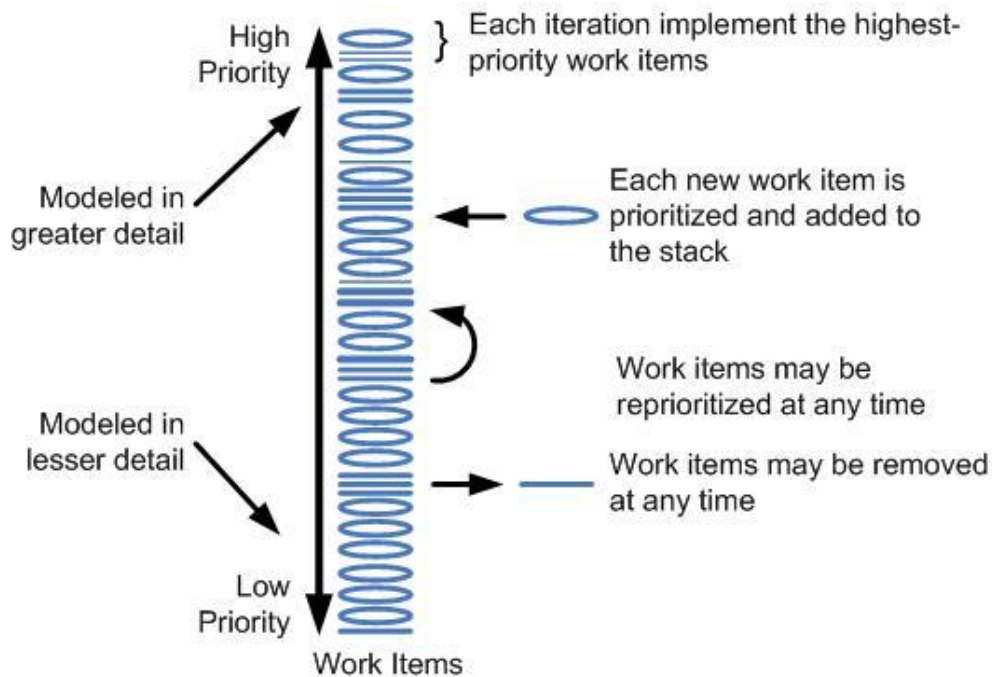


Figure 4. Disciplined agile work management process

Lean Work Management Process

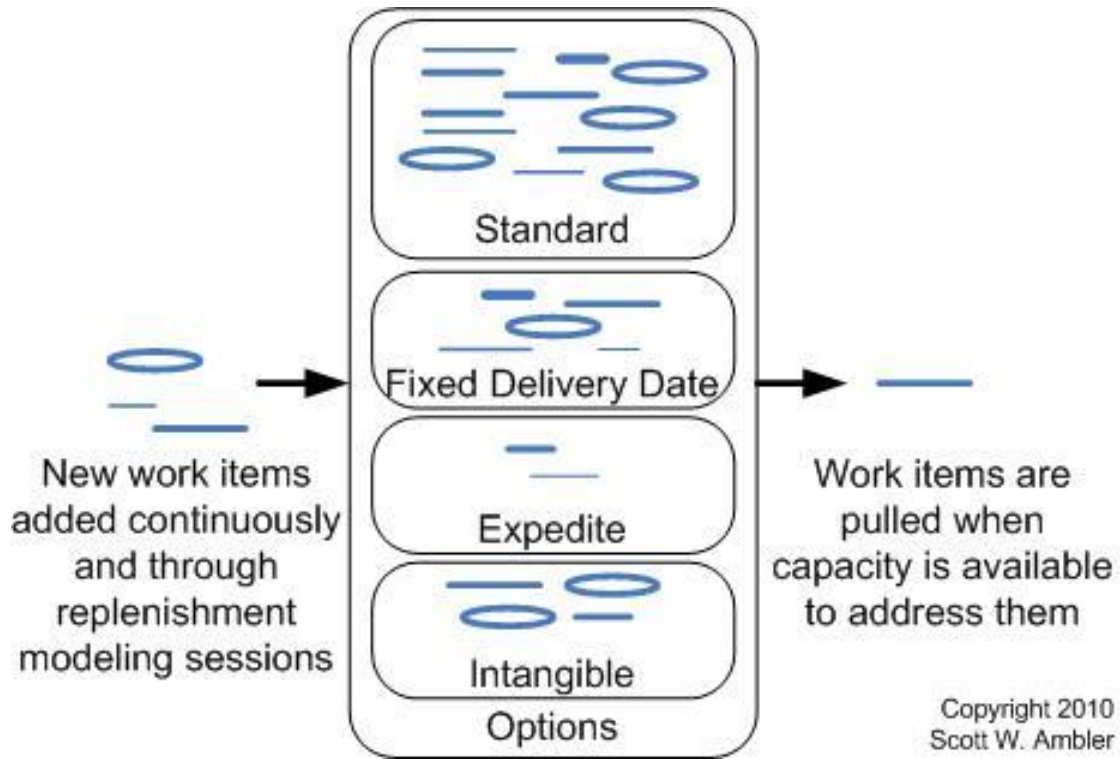


Figure 5. Lean Work Management Process

Work Management in KANBAN

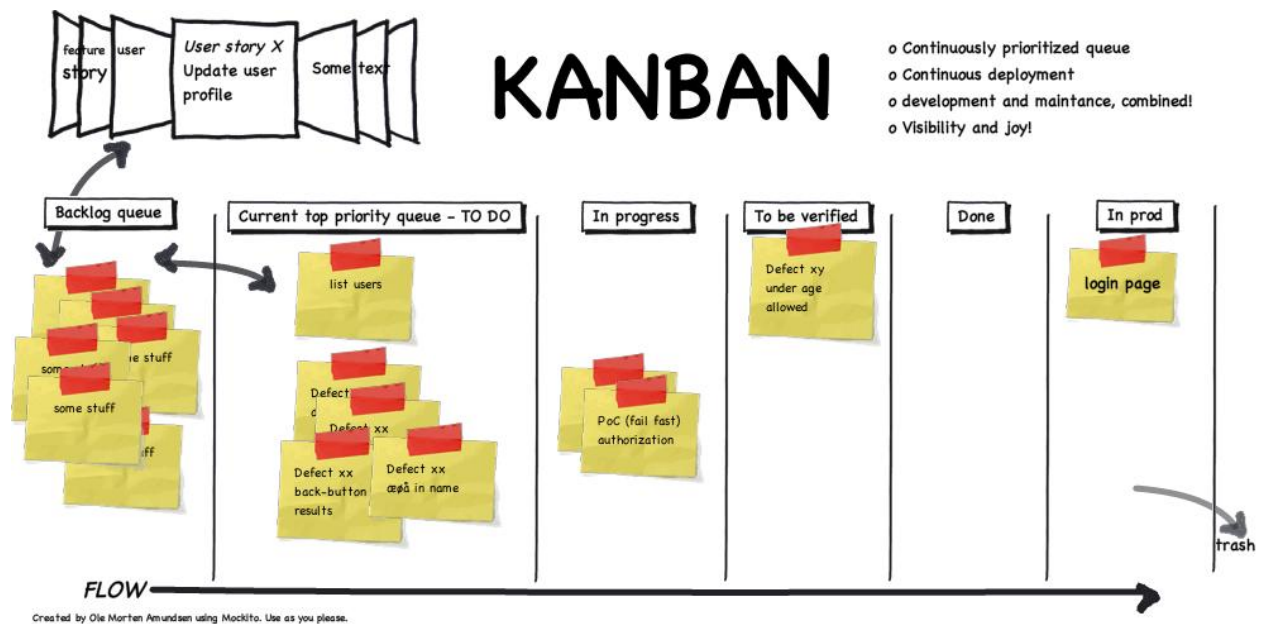
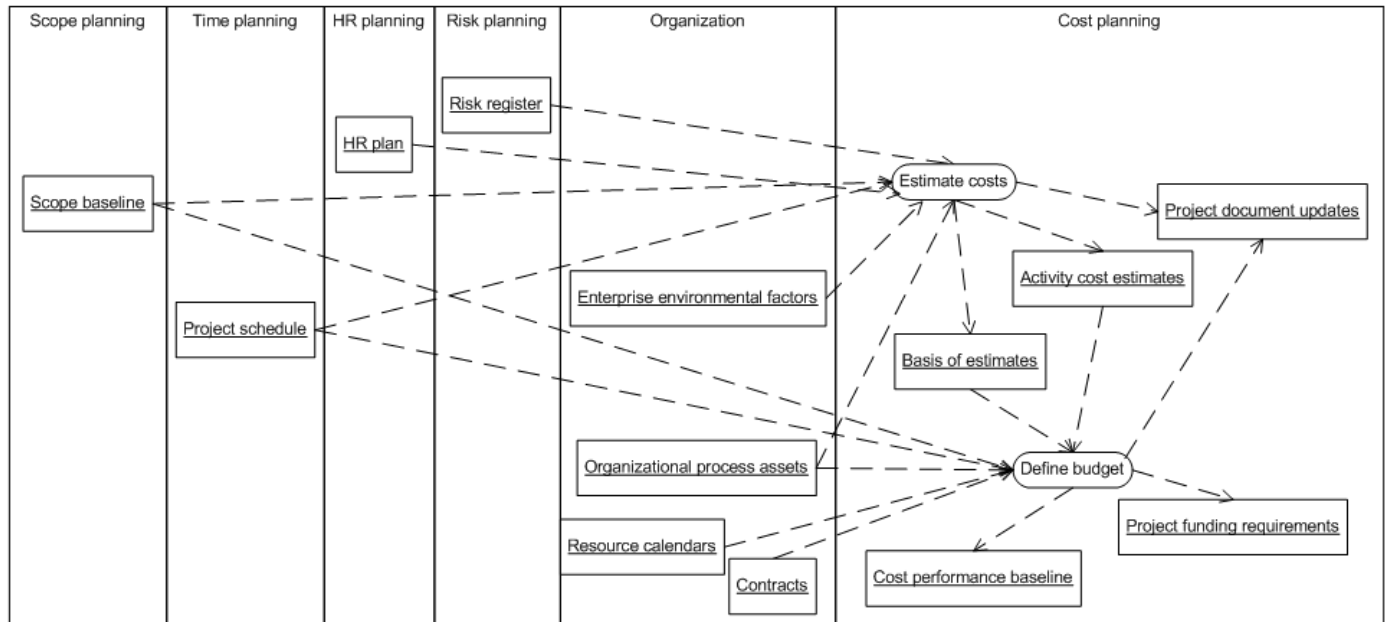


Figure 6. Work management example in KANBAN

## Cost Planning

Includes cost estimation and budget determination. Pictorially expressed on the next figure:



### Cost Estimation by Cost Types

- direct costs – attributed specifically to a detailed or summary activity of the project
- indirect costs - incurred by the project, but the goods or services are also used for non-project activities
- variable costs - fluctuate and can't be predicted with absolute certainty
- fixed costs are static throughout the project or have only a small likelihood of fluctuation

### Cost Estimation Methods

- by analogy – order of magnitude (-25% and +75%)
- resource cost rate (-10% and +25%)
- parametric estimation (-5% and +10%)

Budget determination - cost aggregation by WBS. Every WBS element has its monetary cost

## Used Literature

- Definition of Development Strategy, <http://www.businessdictionary.com/definition/development-strategy.html>
- Project Management Institute, A Guide to the Project Management Body of Knowledge (PMBOK® Guide) - Fourth Edition
- 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](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](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>
- Kenny Rubin, Agile Planning, <http://www.scrumalliance.org/resources/265>
- Scott W. Amber, Governing Agile Software Development, <http://www.ddj.com/architect/202401107>
- Scott W. Amber, Agile Best Practice: Prioritized Requirements, <http://www.agilemodeling.com/essays/prioritizedRequirements.htm>
- Dean Leffingwell, Mastering the Iteration: An Agile White Paper, <http://scalingsoftwareagility.files.wordpress.com/2007/09/mastering-the-iteration-an-agile-white-paper.pdf>
- Mike Cohn, Sources for agile planning and estimating, <http://www.mountangoatsoftware.com>
- Mike Cohn, Estimating With Use Case Points, [http://www.mountangoatsoftware.com/system/article/file/32/Use\\_Case\\_Points.pdf?1267552460](http://www.mountangoatsoftware.com/system/article/file/32/Use_Case_Points.pdf?1267552460)
- Mike Cohn, Selecting the Right Iteration Length for Your Software Development Process, <http://www.mountangoatsoftware.com/articles/30-selecting-the-right-iteration-length>

- Joel Spolsky, Evidence Based Scheduling, <http://www.joelonsoftware.com/items/2007/10/26.html>
- KANBAN example, <http://www.crisp.se/kanban/kanban-example.pdf>
- Ole Morten Amundsen, Kanban and Scrum, combined!, <http://olemortenamundsen.wordpress.com/2010/03/19/kanban-and-scrum-combined/>
- Cost planning template, <http://www.projectmanagementdocs.com/templates/cost-management-plan.html>