



MASARYKOVA UNIVERZITA

## PV213 Enterprise Information Systems in Practice

### 13 – Closing



# MASARYKOVA UNIVERZITA

Tento projekt je spolufinancován Evropským sociálním fondem a státním rozpočtem České republiky.



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

PV213 EIS in Practice: 13 – Closing

Tento projekt je spolufinancován Evropským sociálním fondem a státním rozpočtem České republiky.



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

# Project process



## Example - Reservation system - Motivation

In the company there exist some small equipments (mobile phones, tablets, beamers, flip charts, etc.) which can be used by anyone (pool equipment). Till now reservation of these equipments was done on the paper that is maintained by the secretary. This somehow works but causes several problems. Everyone who wants to make a reservation must visit the secretary and check whether equipment is available at given time. This causes inefficient work time and decreased well-being for employees as well. Another problem is that on the end of the month secretary has to rewrite data about the usage of equipments to the reporting system used by management. These data are then used for analysis whether it is required to buy additional items of given equipment.

## Example - Reservation system - Brief requirements

It is required to create internal tool for reservation of equipments. Definitions of these equipments (name, quantity, description) is stored in external inventory system (SAP) and they are available via web services. Tool must allow to make a reservation of the equipment between specified dates (and times). Only the author of the reservation or special user (e.g. secretary) can finish (or cancel) the reservation. To better indicate real reservation requests it must be possible to mark reservation request even when some equipment is already reserved. Tool must support calendar view (which equipment is reserved for today, tomorrow, etc.) and equipment view (for given equipment show when this equipment is/was reserved). Tool must support export of usage data in given period to the management reporting system. Tool must be available on standard desktop/notebook PCs and on mobile phones as well.

## Requirements analysis

**Requirements analysis (requirements engineering) is process of summarizing all requirements which has to be implemented.**

- ❏ Source of requirements can be requirement description or list of requirements given by customer
- ❏ The most efficient way of requirements clarification is to do several meetings with customer where also motivation for requirement is covered
  - ❏ Sometimes customers don't really know what they want
- ❏ Both functional and non-functional requirements has to be covered
  - ❏ Non functional requirements are often overlooked
- ❏ It is good to group requirements into several groups according to “source” of the requirement
  - ❏ It can help with splitting requirement into functional areas

## Initial modeling of the system I

**Initial modeling of the system is basement for all further work on the project (work for PM, architects, developers, testers, ...).**

- ❏ It helps to understand what has to be implemented (domain of the problem)
  - ❏ You can do it a little bit later if the whole work is clear
- ❏ Identify boundaries of the system (actors are not part of the system)
- ❏ Identify all use cases and create use case models
  - ❏ Important is to identify all “top” use cases as “top” use cases represents bigger amount of work than “sub” use cases
- ❏ Create business domain model to identify all important entities
  - ❏ Business domain model is not detailed but must contain all important relationships between entities and their multiplicities
  - ❏ It is not needed to recognize generalization/specialization relationship in this phase



## Initial modeling of the system II

- ❏ Create all diagrams which are important for understanding problem domain
  - ❏ Activity diagrams for description of processes
  - ❏ Sequence diagrams to understand important sequences
  - ❏ Object diagrams to understand relationships between objects
- ❏ Check models against requirements and try to find if models covers all requirements
- ❏ Check that whole solution is “logic” - even in the requirements there can be some requirements missing
- ❏ Do modeling in several iterations - nothing is perfect on the first try
- ❏ **Check that whole modeled solution will help customer (it solves some real problem)**

# Software development process



## Calculation

- Manufacturing costs
  - personnel
  - HW
  - travelling
  - other
- Risk factor
- Warranty
- Profit

## Estimation

- How could you estimate this project?
  - Expert estimation
    - Work break down
    - Planning poker
    - Affinity estimation
  - Function point analysis
  
- Include “load” factors (review, travels, ...)
- It is just an estimation!

## Team establishment

- To be considered
  - Specialization
  - Experience
  - Costs
  - Personality
  
- Which roles should be defined in our project?

## Risk management

- Main project risk areas
  - Feasibility
  - Available experience
  - Missing expenditure or cost
  - Project management, roles, processes
  - Partners and subcontractors
  - Risk analysis

## Development process selection

- Waterfall?
- Iterative and incremental?
- Agile?
  
- Which one to use?
  - Contracting
  - Customer involvement
  - Requirements
  - Organization
  - Development team and infrastructure
  - Project size
  - Safety and security aspects of the product

## Project management

### ➤ Planning

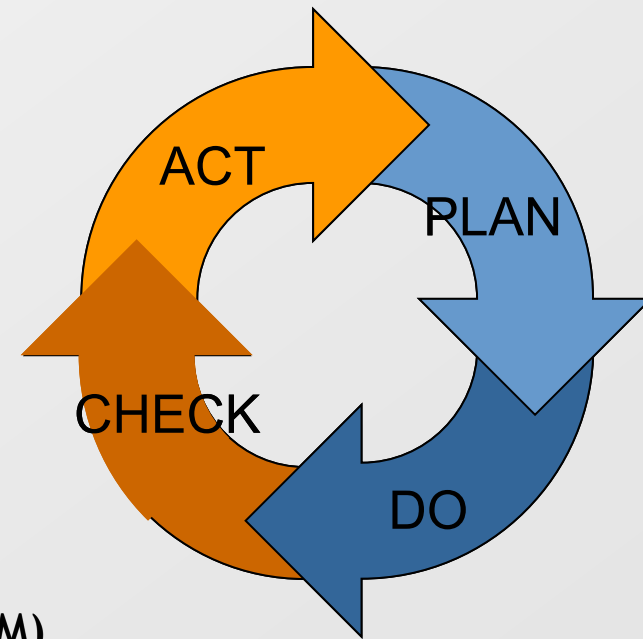
- Define, allocate & manage work packages
- Establish and manage operational schedule
- Initialize & re-estimate the budget
- Project Meeting Schedules & Agenda

### ➤ Realizing

- Project Start-Up
- Constitute and facilitate the project team
- Manage stakeholders & expectations
- Process Change Requests & Defects
- Contribute to support activities (CM, QM, RM)
- Deliver final products to Customer

### ➤ Controlling

- Monitor the project progress, deviations and reporting





## Quality assurance

- ▣ Providing confidence that quality requirements will be fulfilled with respect to
  - ▣ Strategic targets and goals of quality organization
  - ▣ Basic processes
  - ▣ Customer interests
  - ▣ Third parties

## Architecture of the information system I

### Define architecture from the topmost view.

- ❏ Decide which architecture approach is a appropriate for your solution (1-tier, n-tier)
- ❏ Create logical model of all tiers
  - ❏ Define clear boundaries and interfaces
- ❏ Create physical deployment model
- ❏ Decide whether own private infrastructure or some cloud platform will be used
  - ❏ Consider (peak) usage of the system
  - ❏ Consider estimated future usage
  - ❏ Consider security/privacy/legal issues

## Architecture of the information system II

- ❏ From the business domain model create detailed class model(s)
  - ❏ Identify also generalization/specialization relationships
- ❏ Create all models which are important for developers
  - ❏ Object / Component / Package diagrams (structural)
  - ❏ Activity / State machine / Sequence diagrams (behavioral)
- ❏ Check that proposed solution meets functional and non-functional requirements
  - ❏ Check that proposed solution is testable
  - ❏ Check that there aren't build-in security issues
  - ❏ Check if solution can meet expected availability and performance

## Architecture of the information system III

- ❏ Check interfaces to external systems
  - ❏ Consider approaches when external systems are down
- ❏ Define and setup tools and processes for achieving product quality
  - ❏ Choose appropriate programming language / frameworks
  - ❏ Choose appropriate tools for checking code quality
  - ❏ Define code reviews
  - ❏ Execute processes regularly and often
- ❏ Write documentation and guidelines for developers
- ❏ Organize architectural meeting where you present to all members ideas how solution will be build
  - ❏ Do it then even several times during project lifetime

## Configuration management

**Configuration management is supporting role solving processes and tools for successful project execution.**

- ❖ Define tools and processes (configuration management plan)
  - ❖ Version control system
  - ❖ Document management system
  - ❖ Requirement / bug / change tracking system
  - ❖ Integrated development environment
  - ❖ Build system / Continuous integration system
  - ❖ Branching strategy, merging rules
  - ❖ Release processes, rules for versioning
  - ❖ Environments, rules for accessing environments

## Testing I

**Testing is important part of the product development and can represent the same amount of work as coding (or even more for critical systems).**

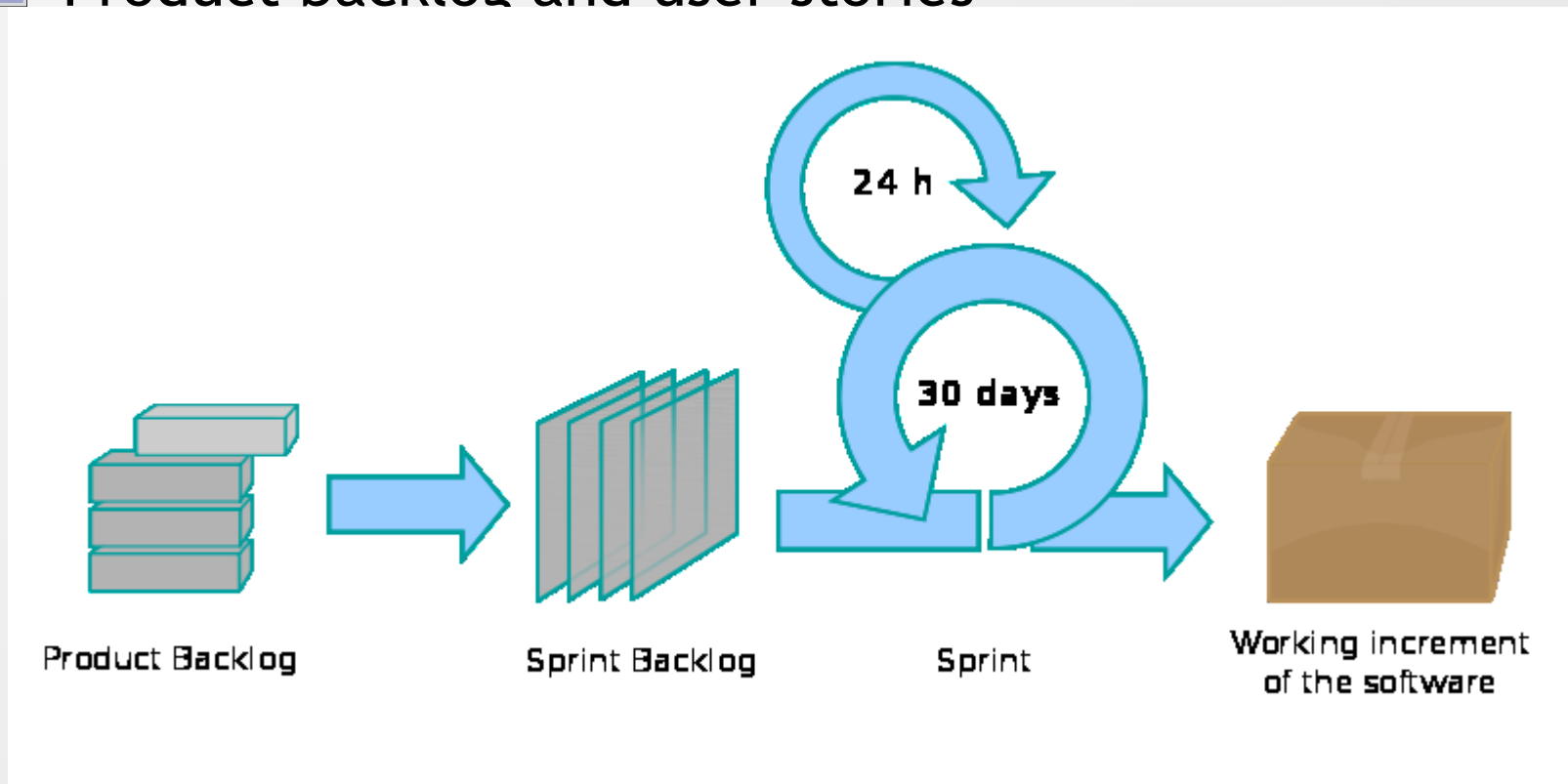
- ❏ Create the test plan for automatic and manual tests
  - ❏ Rules for unit, integration tests, system and acceptance tests
    - ❏ Code coverage, amount of passed tests for acceptance
  - ❏ Not everything can be tested automatically
  - ❏ Define clean testing environment similar to productive environment
  - ❏ Define test cases
- ❏ Consider using test driven development approach (TDD)
  - ❏ Standard in agile development methods
  - ❏ Test driven development allows you to do changes easily without headache what will be destroyed by the change

## Testing II

- ❑ Together with architect define and setup processes and tools for testing static code quality
  - ❑ Code reviews processes
  - ❑ Static code analysis
- ❑ Together with configuration manager define and setup processes and tools for testing
  - ❑ Automatic tests tools and processes
    - ❑ Integration with continuous integration system
  - ❑ Run automatic tests as often as possible (at least once a day)
  - ❑ Use good tools to support manual testing
  - ❑ Generate summary test reports for different testing methods
- ❑ Define test exit criteria - acceptance tests
- ❑ Try to involve customer early in testing phase

## Scrum

### Product backlog and user stories





## Closing the project and maintenance phase

On the end of the main development project is “switched” to the maintenance phase.

- On the end of the project do the experience workshop
- Define the maintenance team
  - Make persons in the team substitutable
- During maintenance carefully monitor new requirements
  - If amount of new (incompatible) requirements exceed acceptable level initiate new negotiations with customer
- If you do the standard product define rules and prices for the product support
- Keep the documentation up to date
- For long term projects try to avoid fluctuation of people
- If you take over the maintenance from other team negotiate that members from the original team will be available for consultations

# Děkuji za pozornost.

Tento projekt je spolufinancován Evropským sociálním fondem a státním rozpočtem České republiky.



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ