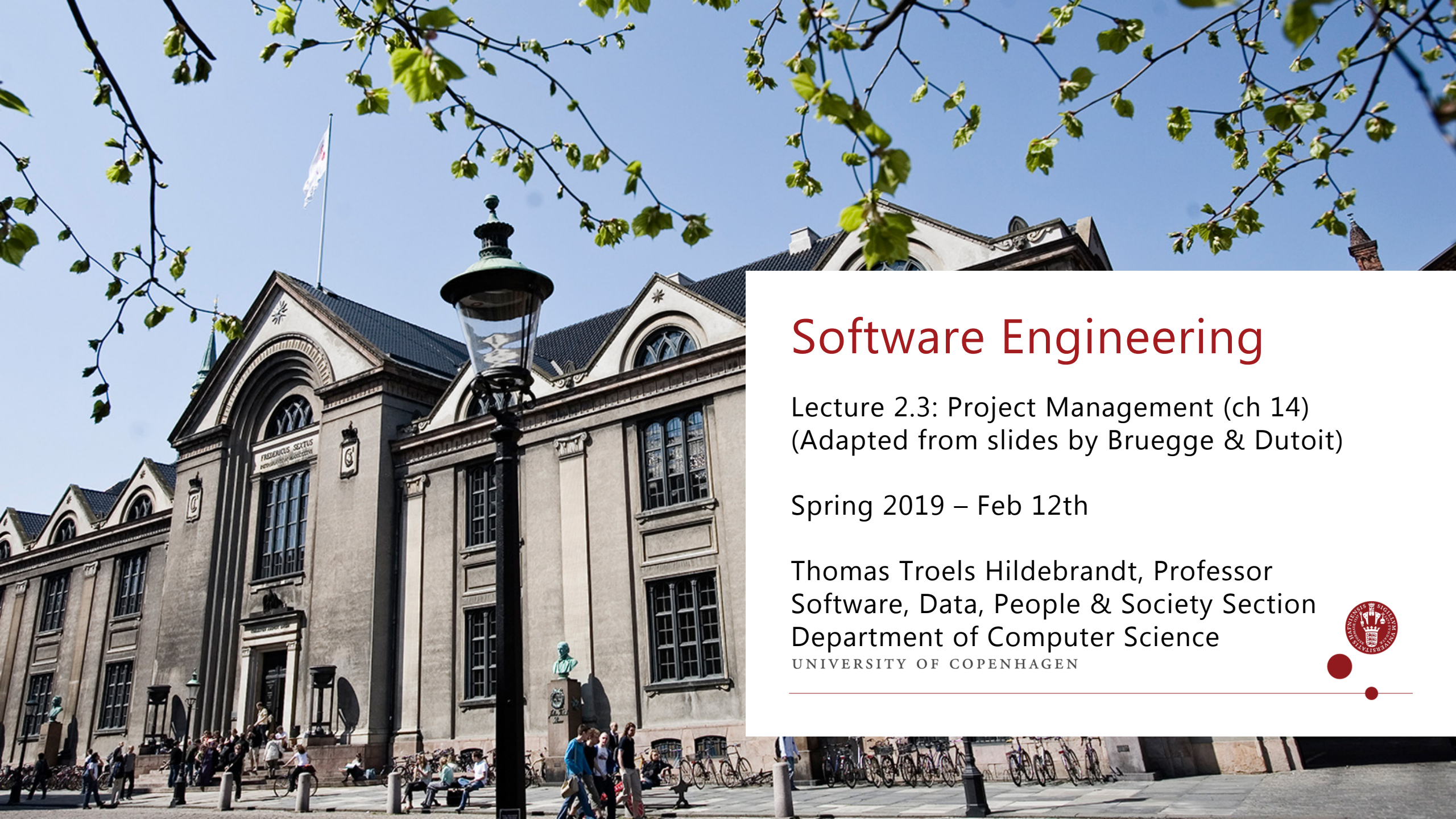


Software Engineering

Lecture 2.3: Project Management (ch 14)
(Adapted from slides by Bruegge & Dutoit)

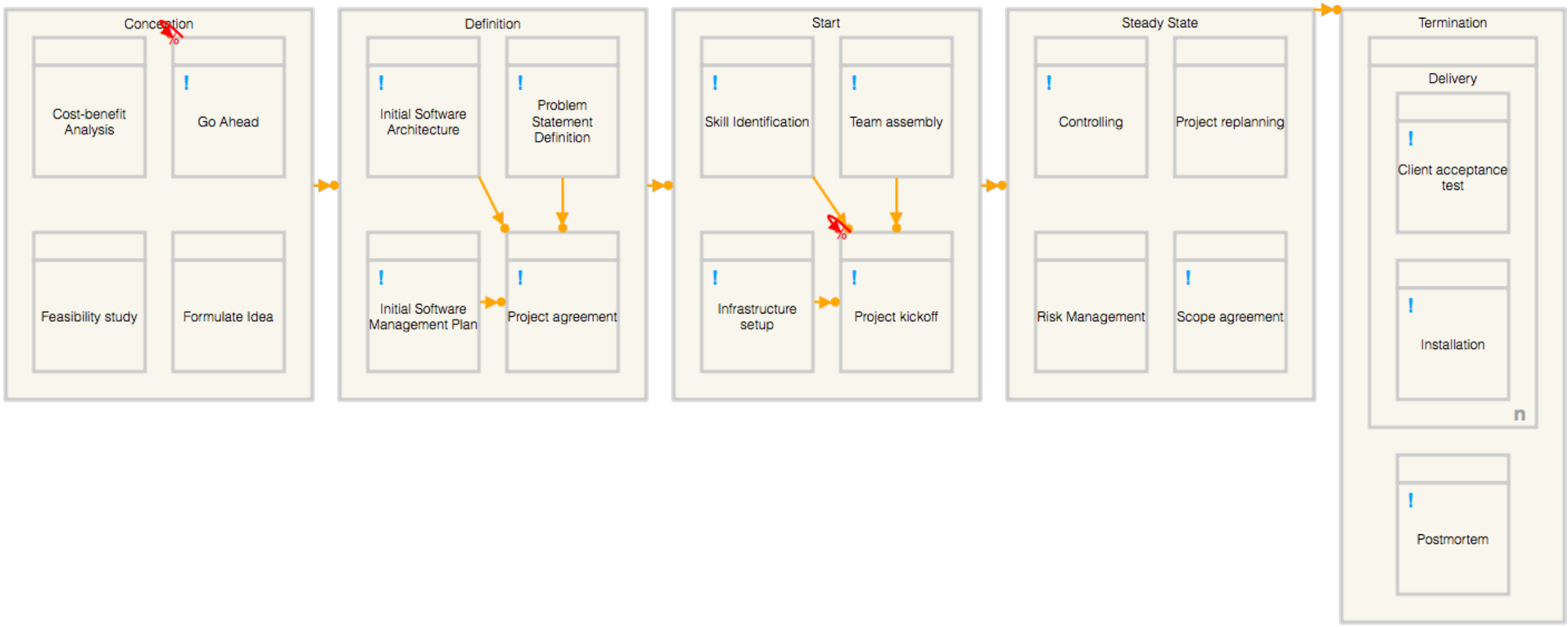
Spring 2019 – Feb 12th

Thomas Troels Hildebrandt, Professor
Software, Data, People & Society Section
Department of Computer Science
UNIVERSITY OF COPENHAGEN

Software Project Management Activities

- Conception
- Planning
  - Speciying results, tasks (description, role, possibly input/output, duration), dependencies, milestones, schedule and estimating resources
- Organizing
  - Identifying roles and responsibilities
- Controlling
  - Monitor the process and deviations, reconfirming expected performance, addressing problems, sharing information, reallocating resources, changing schedule, renegotiate outcome
- Terminating
  - Delivering system to client, reviewing project history to extract lessons learned, modifying templates

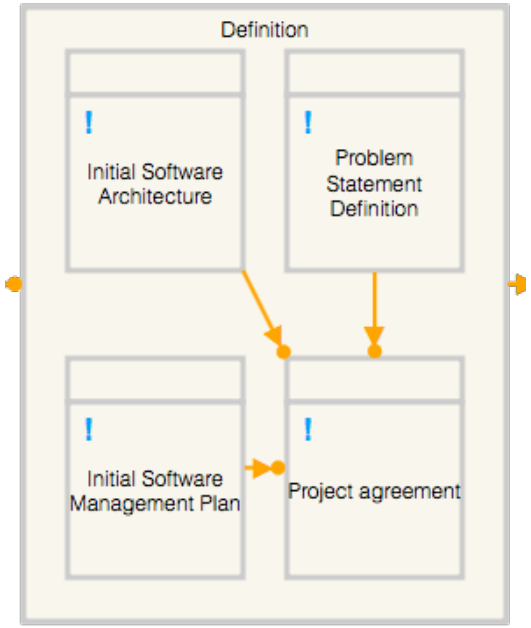
Software Project Management Activities DCR Graph



http://www.dcrgraphs.net/Tool?id=9210#

Planning the Project – work products

- Problem statement
- Top-level design (initial software architecture)
  - Subsystems and functionality
- Initial Software Project Management Plan (SPMP)
  - Work break down structure
  - Schedule
  - Work packages
  - Budget



Problem statement

- Application & Problem domain
- Scenarios
- Functional requirements
- Non-functional requirements
- Target environment
- Deliverables and deadlines

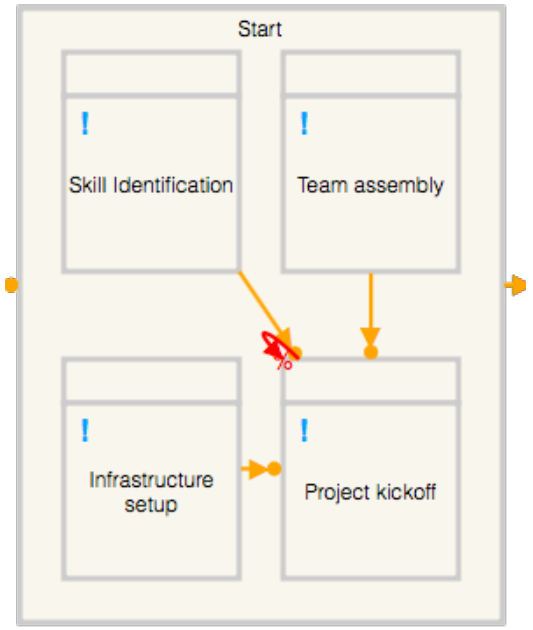
Top-level design (initial architecture)

- Main sub systems and their functionality – not interfaces
- Serves as basis of planning the organisational units and communication
- Will be refined during the System Design activity later.

Software Project Management Plan

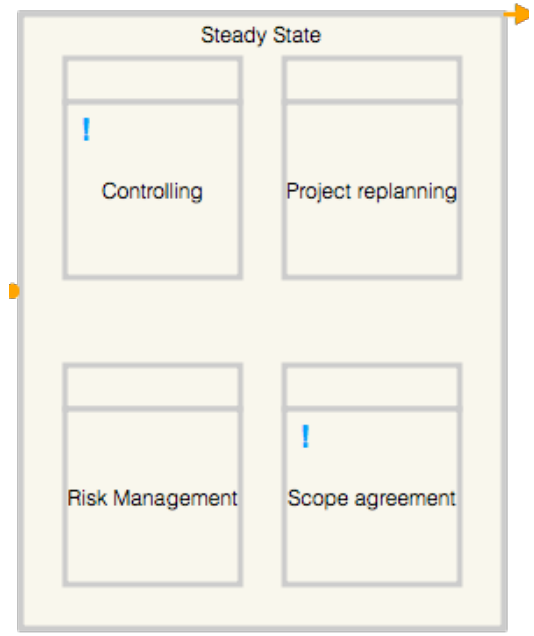
- Work breakdown structure (WBS)
  - functional decomposition (Authentication, Information flow security, ..)
  - subsystem decomposition (UI, DB, Control)
  - Geographical (Denmark, Pakistan, ⋯)
  - Organisational (Marketing, Operations, Testing⋯)
  - Seek inspiration from previous WBS
  - Involve key developers
  - Identify work gaps and overlaps
    - Any work not mapped to tasks? Same taks included in several activities?
- Initial Schedule
  - Only details for the first month

Project Start – Activities and work products



- Infrastructure setup
  - Scheduled (milestones, reviews, status meetings)
  - Event-based (email, project management systems)
- Skill identification
  - Application domain, communication, technical, quality, management
- Team assembly
  - Management and technical roles
  - Team sizes – ideal around 5-6
  - Flat or gradual staffing
- Project agreement (baseline for client acceptance)
- Kick-off meeting

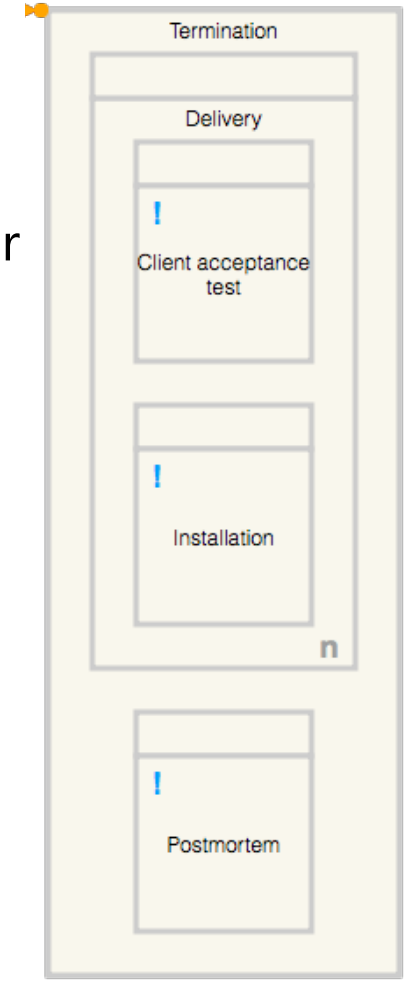
Steady state: Controlling the project



- Collecting status information
  - Meetings
  - Sharp milestones
  - Project reviews
  - Code inspections
  - Prototype demonstrations
- Metrics
  - Deviation from schedule, resources expended, changes in project group, changes to system (stability), financial status, technical progress (size and quality), modularity (breakage), maturity (mean time between failures)
- Risk management
  - Managerial or technical, likelihood & impact, mitigation strategy

Terminating the project

- Prepare client acceptance test
  - Present system and compare to acceptance crieteria in project agreeement, discuss subsequent maintenance, knowledge transfer & enhancements
- Manage system integration, testing, installation, training
  - Be careful with big-bang installation and insuficient training
- Post mortem (lessons learned)
  - Planned vs. Actual delivery dates, number of defects, technical and managerial problems, suggestions for future projects



Agile project management: Scrum [Schwaber & Beedle, 2002]

- Small project increments in short time bursts (**Sprints,** typically 30 days)
- **Potentially deliverable product increment**
- **Product backlog**:
  - Prioritized list of requirements (starts as business requirements, graduately refined into system requirements)
  - Current understanding, not fixed contract
  - Grouped into releases with planned release dates
- **Spring planning meeting**
  - Select **sprint backlog** (tasks and effort estimates)
- **End of sprint**
  - Product increment?, Project finished? Adjust release planning?

Organising the Agile Scrum Project

- **Product owner**
  - Responsible for product backlog, not the team during the sprint
- **Scrum Master**
  - Sets up rules, manages the daily Scrum meetings, monitors progress. Interface betwen product owner and Scrum Team
- **Scrum Team**
  - Develops the project increments. Cross functional and no strict roles.

Controlling the Scrum Project

- At **daily Scrum meeting** (max 15 minutes) each team member reports:
  - Status: Progress since last daily meeting
  - New issues: What prevents progress
  - New actions: work promised for the rest of the day

- **Burn down chart**
  - Estimated remaining time for each item in the sprint, updated daily

Exercises

- DCR Solutions has a team of programmers in Pakistan, a CEO with background as computer scientist, a head of sales and a team of computer science researchers in Denmark also able to work part-time as consultants. The client municipality has its own programmers with knowledge of interfaces of local systems and case workers with knowledge of the local case workflows and the DCR design portal.
- The OCM is divided in the following sub systems: 1) A process engine server implementing the logic of the DCR process language invented and continuously developed by the CEO and the researchers. 2) A GUI Client. 3) An integration with the local document management system used in the municipality, 4) A formalisation of the case work flows as DCR graphs
- How will you allocate roles to the development of the OCM ? What are the issues, proposals, arguments, criteria and resolution?