Project Planning and Control

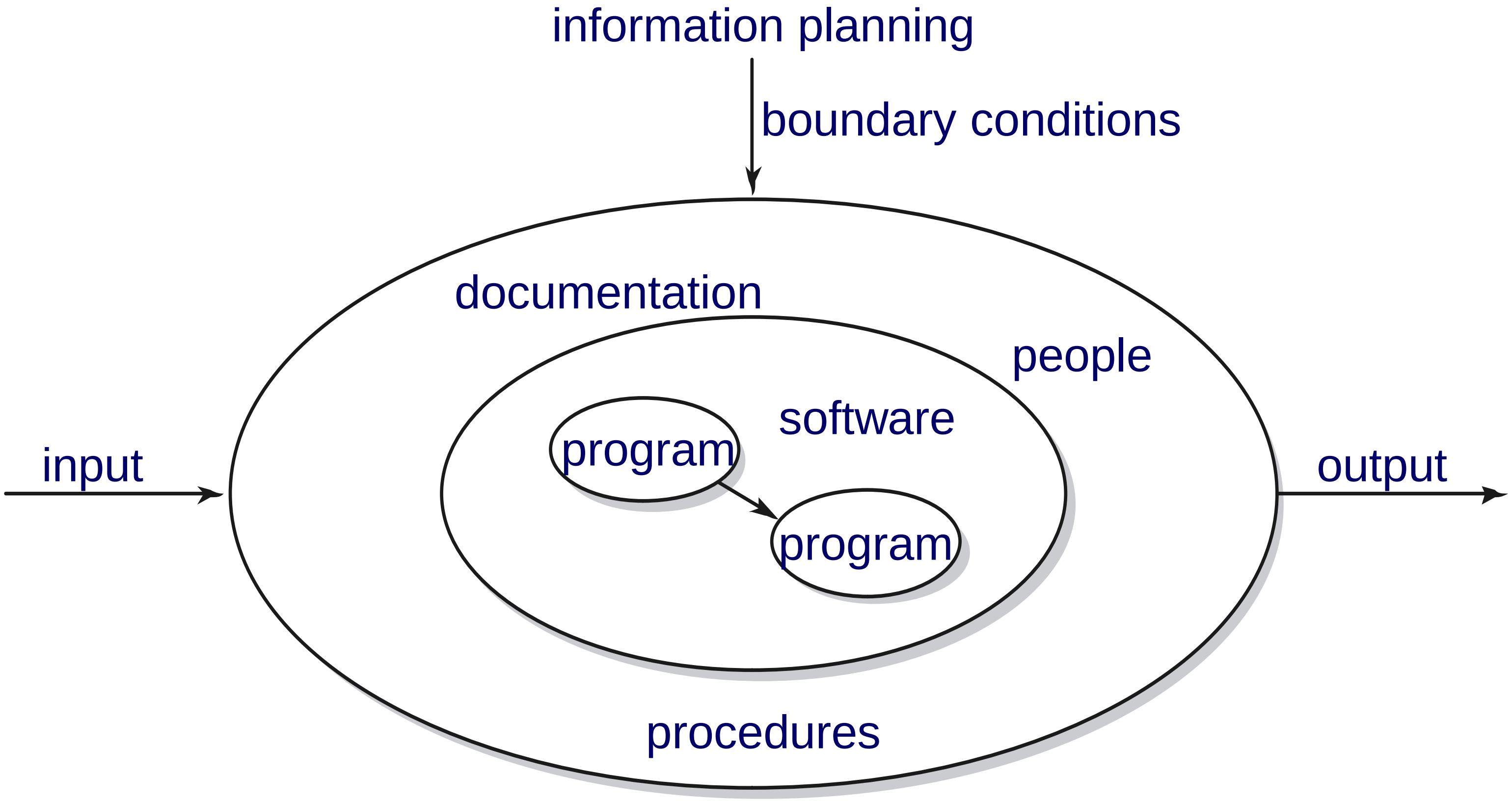
Main issues:

- How to plan a project?
- How to control it?

The systems view on software development

systems view: input → system → output

system in this lecture: a *software* system in a broad perspective



Example: information plan for university registration of student data

- Relations to other systems: personal data, courses, course results, alumni, …
- Use both by central administration, at faculty level, and possibly by students themselves
- Requires training courses for administrative personnel
- Authorization/security procedures
- Auditing procedures
- External links, e.g. to scholarship funding agencies, ministry of education

Contents of a plan for a single project (1)

- Introduction
- Life cycle model
- Organization of project (e.g. team; chpt. 5)
- Standards, guidelines, procedures (e.g. on documentation)
- Management activities
- Risks (availability of resources; chpt. 8)
- Staffing

Contents of a plan for a single project (2)

- Methods and techniques (e.g. design, implementation or testing)
- Quality assurance (chpt. 6)
- Work packages – work breakdown structure
- Resources (see also: risks)
- Budget and schedule (chpt. 7)
- Changes – procedures how to handle them
- Delivery

Project control

- **Time,** both the number of man-months and the schedule
- **Information,** mostly the documentation
- **Organization,** people and team aspects
- **Quality,** not an add-on feature; it has to be built in
- **Money,** largely personnel

Systems view of project control

- **Irregular variables: cannot be controlled by PM**
  - E.g. experience of the user
- **Goal variables: things one wants to achieve**
  - e.g. minimize downtime or cost, or maximize quality
- **Control variables: can be varied by PM**
  - e.g. project staffing, or tools to be used

- **Distribution of variables over categories is not rigid**
  - E.g. cost: *goal* or *control* variable; staffing: *control* or *irregular*
- **To control the project, you have to know the category of each variable, and more (see next slide)**

Systems view of project control: conditions

- **The PM must**
  - know the goals of the system
  - have sufficient control variety
  - have information on state, input and output of the system
  - have a conceptual control model:
    knowledge of how and to what extent the different variables depend on and influence each other.

    In practice this knowledge is not, or only partially available.

Classes of project characteristics

- **Product, process, and resource characteristics**
- **We are interested in the degree of certainty (low/high)**
- **Product certainty:**
  - Clear requirements, known upfront => product certainty is high
  - User requirements change frequently => product certainty is low
- **Process certainty:**
  - E.g. use of unknown tools => process certainty is low
- **Resource certainty:**
  - Depends on availability of appropriately qualified personnel

Invalid combinations of project characteristics

Certainty of				
Product	low	low	low	high
Process	low	high	high	low
Resource	high	low	high	high

Archetypical control situations

Certainty of	*Realization*	*Allocation*	*Design*	*Exploration*
Product	high	high	high	low
Process	high	high	low	low
Resource	high	low	low	low

- **Realization problem:** Ideal situation, just make sure work gets done
- **Allocation problem:** Major issue: controlling capacity
- **Design problem:** How to design the project (milestones, personnel, etc.)
- **Exploration problem:** Major issue: get commitment of all people involved

Control aspects taken into account for the different control situations

- Primary goal for controlling the project
- Coordination/management style
- Software development strategy
- Cost estimation

Control situation: realization

(prod,proc, res)=(high,high,high)

- **Primary goal in control:**
  - Optimize resource usage, efficiency and schedule
- **Coordination/management style:**
  - Standardization of product, process and resources; hierarchy; separation style
- **Development strategy:**
  - Waterfall
- **Cost estimation:**
  - Models (chpt. 7), guard the process

Control situation: allocation

(prod,proc, res)=(high,high,low)

- **Primary goal in control:**
  - Acquisition & training of personnel
- **Coordination/management style:**
  - Standardization of product and process
- **Development strategy:**
  - Waterfall
- **Cost estimation:**
  - Models, expert estimates, sensitivity analysis

Control situation: design

(prod,proc, res)=(high,low,low)

- **Primary goal in control:**
  - Control of process
- **Coordination/management style:**
  - Standardization of process
- **Development strategy:**
  - Incremental
- **Cost estimation:**
  - Expert, sensitivity analysis

Control situation: exploration

(prod,proc, res)=(low,low,low)

- **Primary goal in control:**
  - Maximize results (functionality and/or quality), lower risks
- **Coordination/management style:**
  - Mutual adjustment (adhocracy), commitment, relation style
- **Development strategy:**
  - Incremental, prototyping, agile
- **Cost estimation:**
  - Agile, risk analysis, provide guidance

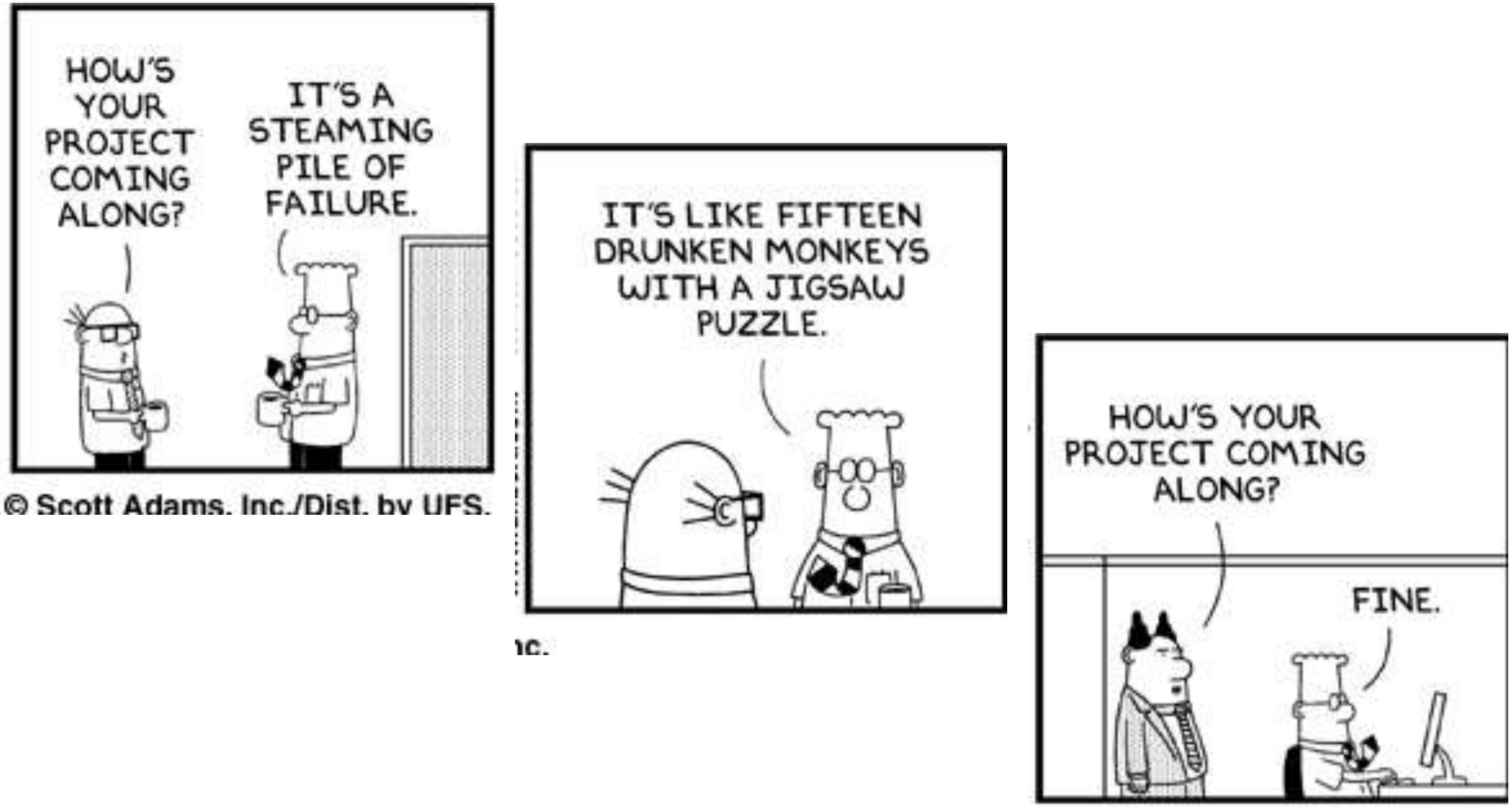
Risk management

- **Risk management is project management for adults (Tim Lister)**
- **In software development, we tend to ignore risks:**
  - We’ll solve the problem on time
  - Requirements will be stable
  - No one will leave the project
  - …

Definitions of risk

- **effect of *uncertainty* on *objectives* (ISO)**
- **the probability of *uncertain* future *events* (Open Group)**
- **p x s, where**
  - p = the probability of a hazard (resulting in an adverse event)
  - s = the severity of the event

Risks evaporate in higher echelons


HOW'S YOUR PROJECT COMING ALONG?
IT'S A STEAMING PILE OF FAILURE.
IT'S LIKE FIFTEEN DRUNKEN MONKEYS WITH A JIGSAW PUZZLE.
HOW'S YOUR PROJECT COMING ALONG?
FINE.

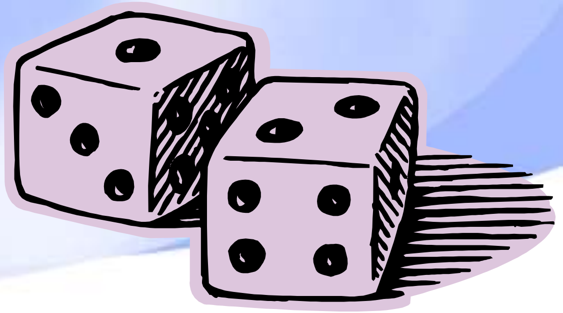
© Scott Adams, Inc./Dist. by UFS.

nc.

Top ten risk factors (Boehm, 1989)

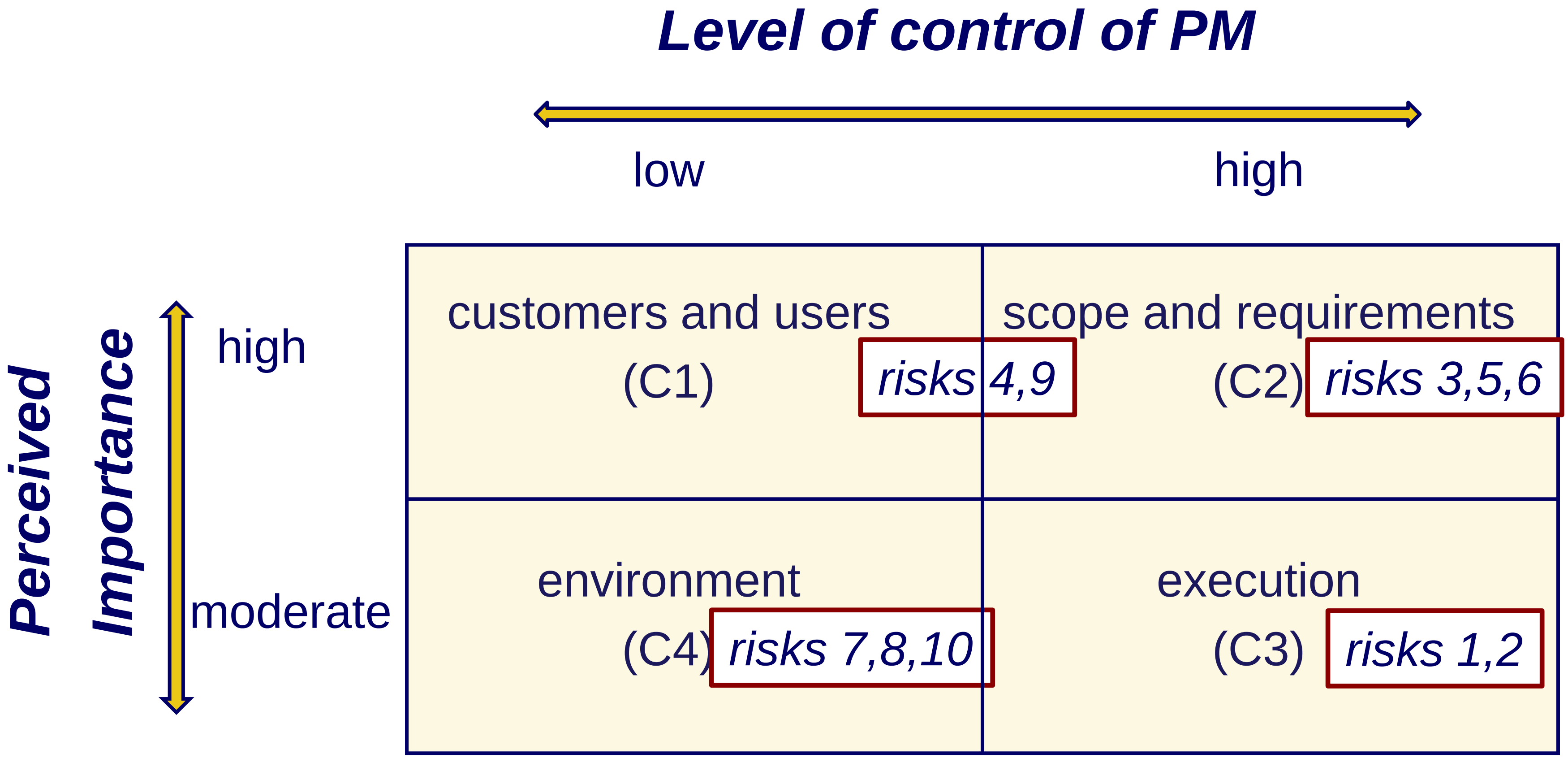
1. Personnel shortfall
2. Unrealistic schedule/budget
3. Wrong functionality
4. Wrong user interface
5. Gold plating
6. Requirements volatility
7. Bad external components
8. Bad external tasks
9. Real-time shortfalls
10. Capability shortfalls

Risk management strategy



1. Identify risk factors
2. Determine risk exposure (probability * effect)
3. Develop strategies to mitigate risks
   - Avoid, transfer, or accept
4. Handle risks

Categories of risks (Wallace & Keil, 2004)



Level of control of PM

Perceived Importance

	low	high
high	customers and users (C1) *risks 4,9*	scope and requirements (C2) *risks 3,5,6*
moderate	environment (C4) *risks 7,8,10*	execution (C3) *risks 1,2*

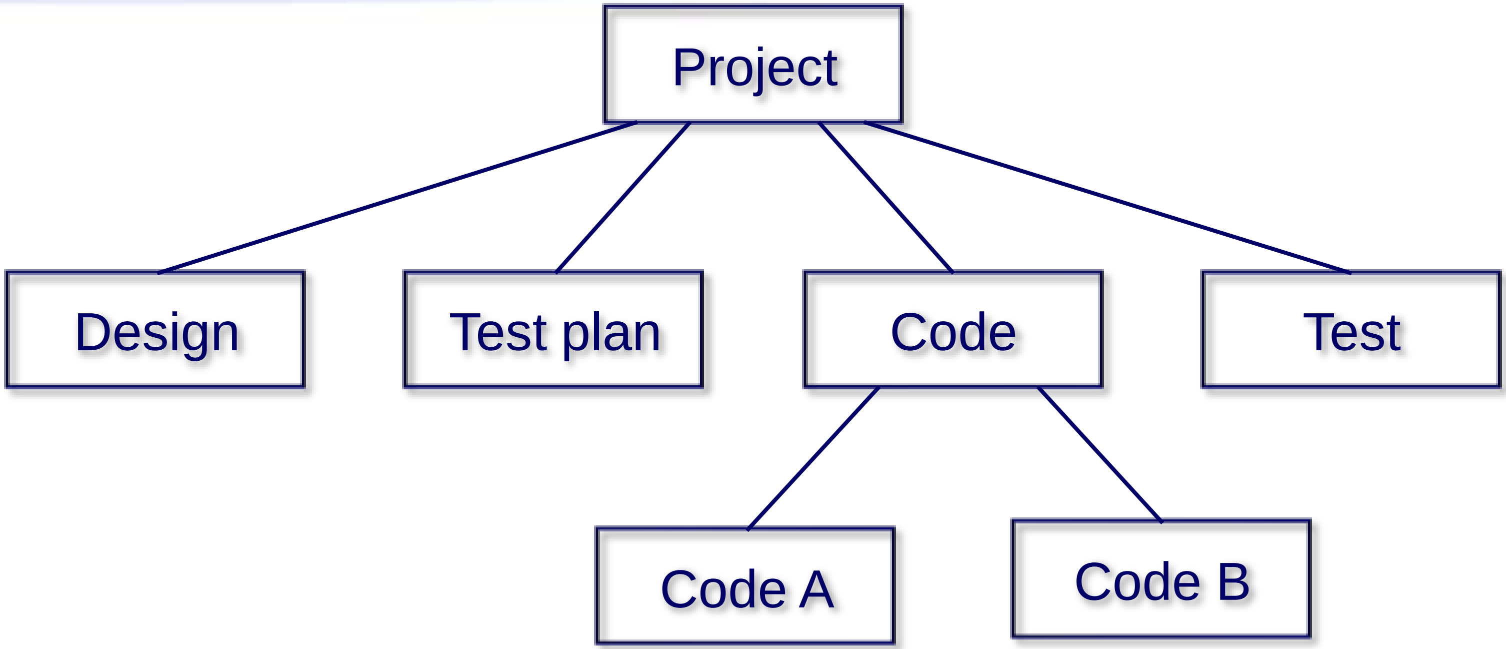
Order of handling: first C3, then C2, then C1 and C4

Techniques for planning and controlling the activities within a project

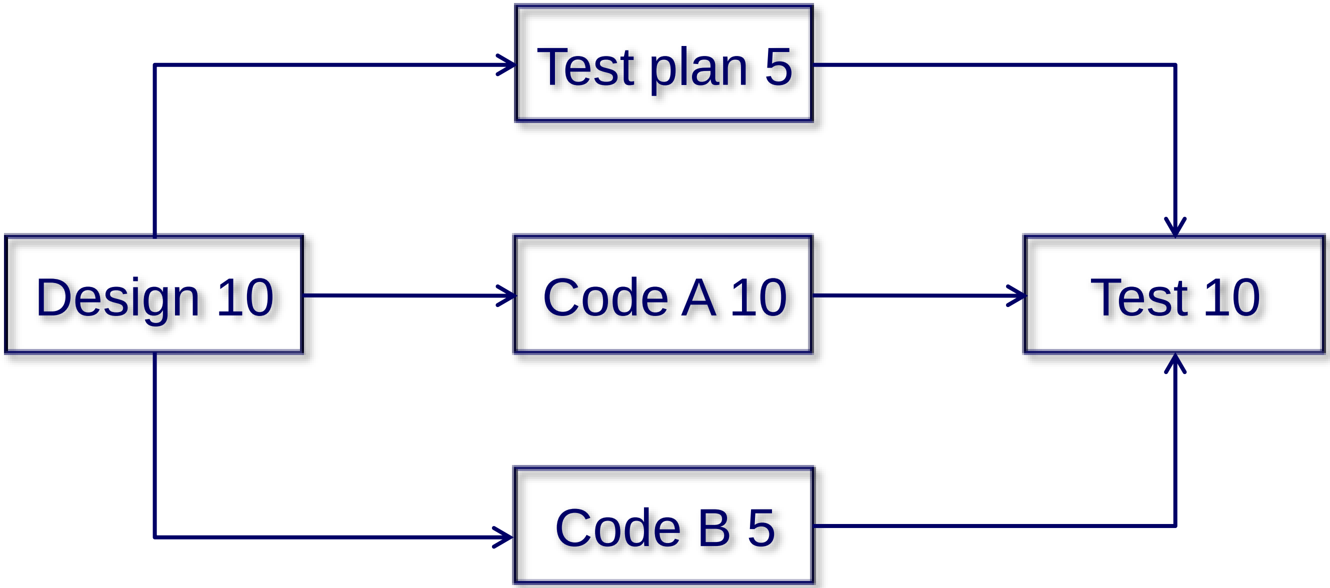
- Work breakdown structure (WBS)
- PERT chart
- Gantt chart

- Agile planning and control

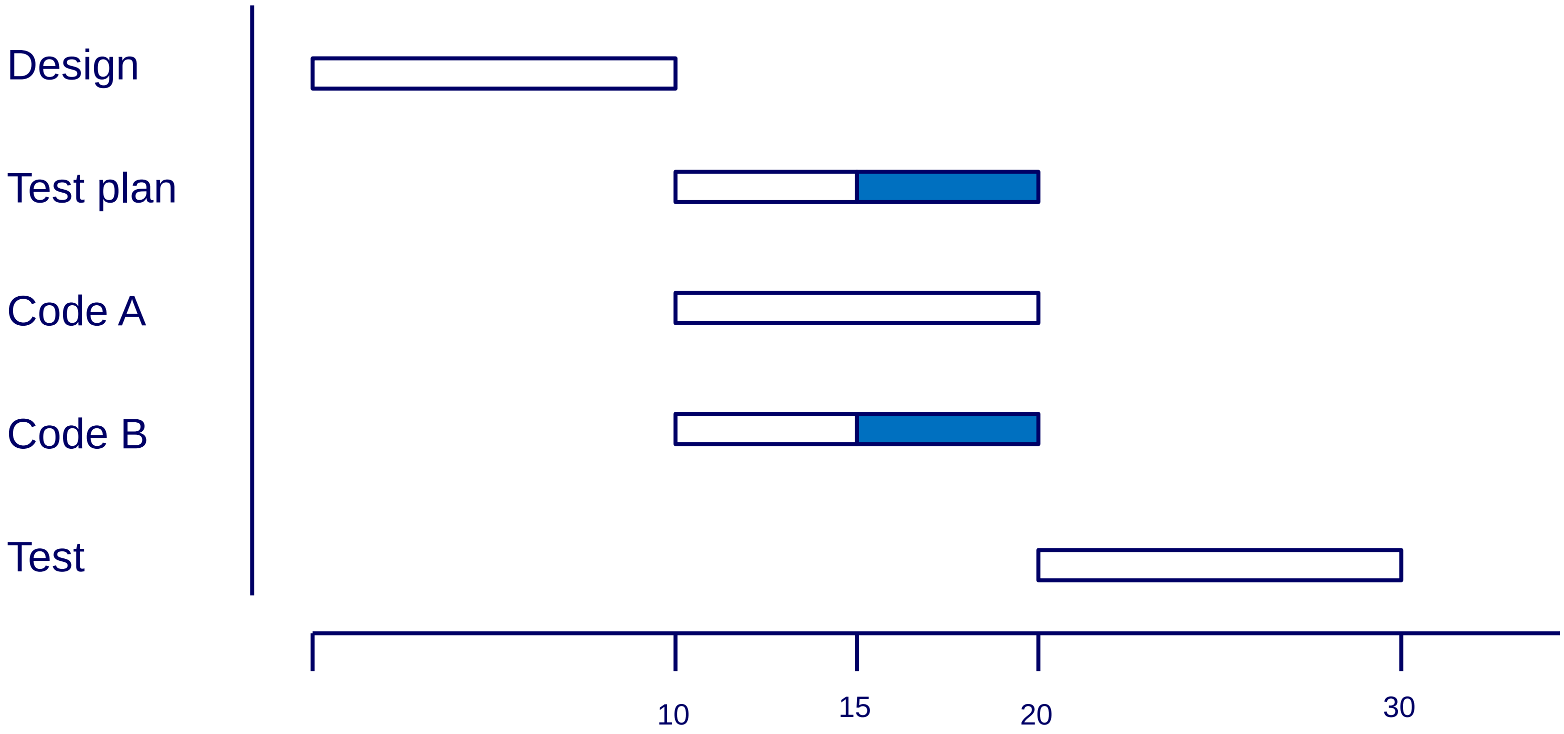
Work Breakdown Structure



PERT chart

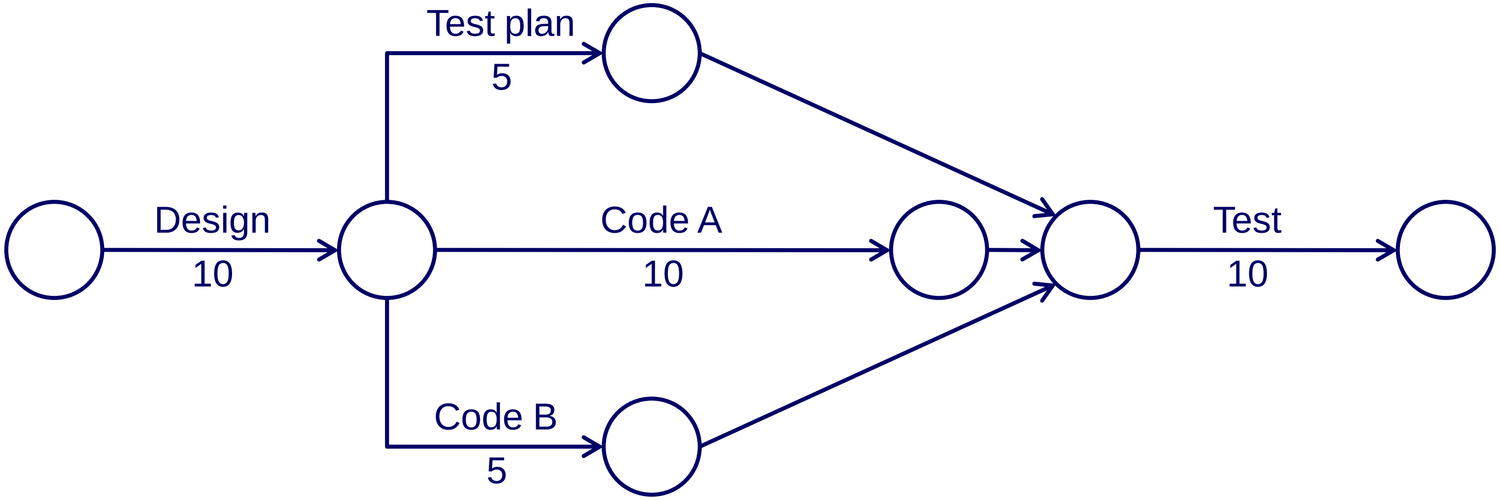


Gantt chart



Zie ook MSO, slide 20 van
http://www.cs.uu.nl/docs/vakken/mso/1011/slides/UMLUP.pptx

Activity-on-arrow network[1)]



Note the dummy activities, needed for synchronization.

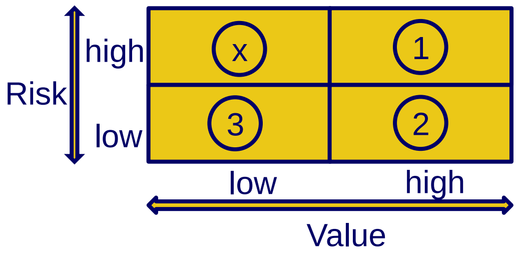
[1)] *As opposed to “activity-on-node network”*

Why task-oriented planning is problematic

- **Activities never finish early**
  - Parkinson’s law: work fills the time available
- **Lateness is passed down the schedule**
  - If either design or coding is late, subsequent testing will be late
- **Tasks are not independent**
  - If design takes more time, so will implementation

Agile planning factors

- **Estimate value of features**
  - e.g. the MoSCoW way
- **Cost of implementing features**
  - Cost of doing it now versus cost of doing it later
- **New knowledge acquired**
  - First do features that bring a lot of new knowledge
- **Risk removed by implementing feature**
  - First *high-value – high-risk*, then *high-value – low-risk* features, finally *low-risk – low-value* features;
  - Avoid *low-value – high-risk* features


Risk
high
low
x
1
3
2
low
high
Value