Software Project Management – summary

Contents

2 Project management foundations

- 项目越大越容易失败，agile 比 water fall 要少失败
- Failure factors of a project:
  1.Incomplete Requirements
  2.Lack of User Involvement
  3.Lack of Resources
  4.Unrealistic Expectations
  5.Lack of Executive Support
  6.Changing Requirements & Specifications
  7.Lack of Planning
  8.Didn't Need It Any Longer
  9.Lack of IT Management
  10. Technology Illiteracy
- Two definitions of project:
  1.A project is a temporary endeavor undertaken to create a unique product or service
  2.Any series of activities and tasks that:
    a) have a specific objective to be completed within certain specifications

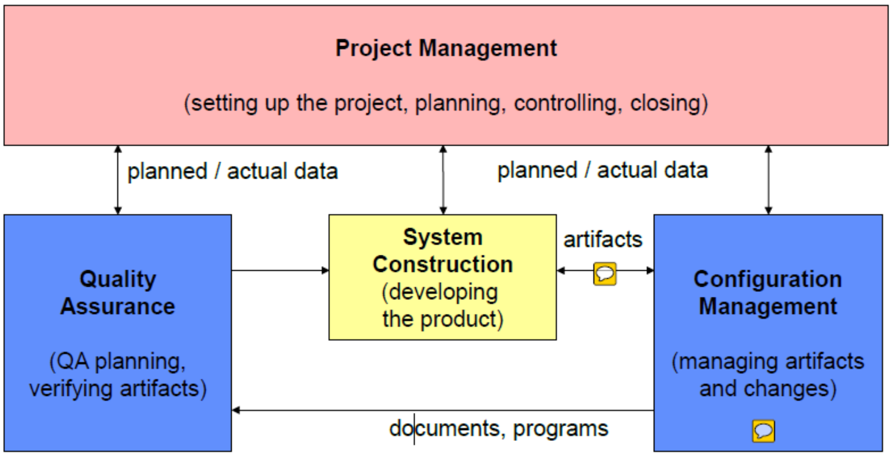
    b) have defined start and end dates
    c) have funding limits (if applicable)
    d) consume resources (i.e., money, people, equipment)

- Operation: ongoing and repetitive
- Project can be the development of a product, operation can be the corresponding production of the product after development
- Applications are neither projects nor operations, but products
- Characteristics of projects:
    a) Temporary endeavor: start and end date
    b) Unique product or service
    c) Performed by people
    d) Constrained by limited resources: budget, time, staff
    e) Planned, executed, and controlled
    f) Have their own organization
- 3 kinds of projects

  Order project:
    a) The development organization develops according to the requirements of an external client.
    b) A contract defines deliverables and donations
    c) Conflicts are typically decided in court
    d) Example: Customer information system

  Internal development project:
    a) Client and developers are with the same organization
    b) The system is typically used internally
    c) Conflicts are solved by the common top-level manager
    d) Example: Portfolio management system of a bank

  Product development project:
    a) Client is often the internal marketing division
    b) The developed product is offered and sold by the company
    c) The money is drawn from the global development budget
    d) Example: A navigation system for mobile devices
- Projects are often a mixture of these
- Software development project


Project Management
(setting up the project, planning, controlling, closing)
planned / actual data
planned / actual data
Quality Assurance
(QA planning, verifying artifacts)
System Construction (developing the product)
artifacts
Configuration Management
(managing artifacts and changes)
documents, programs

就是 QA, construction, configuration(a collection of formal documented procedures)三者都互相连着，而 pm 和他们都有双向联系

- 3 definition of Project management:
    1. Task oriented: A system of procedures, practices, technologies, and know-how that provides the planning, organizing, staffing, directing, and controlling necessary to successfully manage an engineering project.
    2. Goal oriented: Concerned with activities involved in ensuring that software is delivered on time and within budget and in accordance with the requirements of the organizations developing and procuring the software.
    3. Normal: Is the application of knowledge, skills, tools, and techniques to project activities to meet project requirements
- 9 knowledge areas for PMI: (+ management in the end)
    1. Integration
    2. Cost
    3. Communications
    4. Scope
    5. Quality
    6. Risk
    7. Time
    8. Human resource
    9. Procurement (采购)
- Tasks of project management:
    1. Project organization
        a. Defining the roles
        b. Allocating authority and responsibility
        c. Setting up communication structure of the project
    2. Team management
        a. Selecting people
        b. Building the team
        c. Promoting people
        d. Motivating people
        e. Solving conflicts
        f. Releasing the team
    3. Project planning
        a. Defining the goals
        b. Analyzing the environment
        c. Structuring the tasks/activities to be performed
        d. Planning quality issues
        e. Preparing the schedule
        f. Planning resources
        g. Planning cost and budget
        h. Planning risk management
    4. Project monitoring and controlling
        a. Defining measures to control quality, deadlines, resources, and cost

        b. Measuring the progress of the project
    5. Stakeholder management
        a. Identify the stakeholders
        b. Analyze: interests, standpoint (promotor, supporter, endure 忍耐的人, opponent), personality, organizational context
        c. Define actions to be taken
        d. Monitor interests, standpoint, and context
    6. Risk management
        a. Identifying risks
        b. Evaluating risks
        c. Defining mitigation 缓和 and contingency 应急 measures

- Prerequisites for successful Project management
    1. Procedures to plan activities, schedule etc.
    2. Procedures to estimate cost
    3. Procedures to measure the progress of the project
    4. Understanding of development processes
    5. Realistic estimation of own abilities: technical, organizational
    6. Experienced project managers
    7. Appropriate environmental conditions
- Skills of a project manager

  Technical abilities
    1. economical know-how
    2. application know-how
    3. software engineering know-how

  Project management abilities
    1. planning, estimation, controlling
    2. time management, negotiation abilities

  Leadership abilities
    1. communication, decision, delegation
    2. motivation, moderation

  Personal abilities
    1. responsibility, toughness, team spirit
    2. patience, ability to communicate, fairness
- project management is a prerequisite to run a project successfully
- project management is a continuous activity throughout the project

3 Software development processes – overview

- Process: A process is a series of actions bringing about a result.
- Software development process (user needs → software product): 5 activities
    1. translating **user need**s into **software requirements**,
    2. transforming the **software requirements** into **design**,
    3. implementing the **design** in **code**,

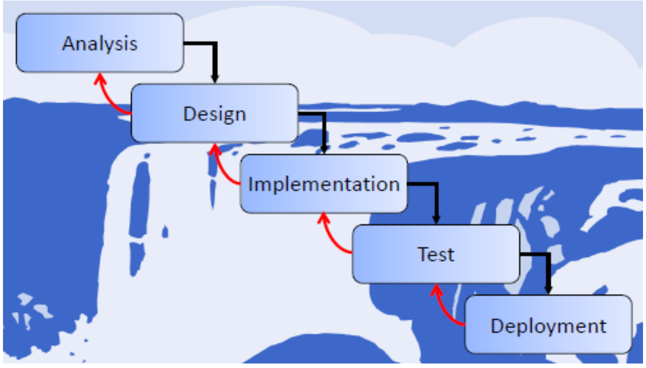
4. testing the **code**,
5. and sometimes, **installing and checking out the software** for operational use

- The 5 above activities may overlap or be performed iteratively
- Software process model:
  an abstraction of similar software processes
  a blueprint (a pattern) used to organize and perform a software process

  A model should be:
  1. useful and helpful
  2. clearly defined
  3. accepted by the development organization
  4. Tool supported

  Goal: reduction of project risks
- Well known process model below:
- Code and fix: coding and error correction together with ad-hoc tests are the only activities performed consciously
  Problems:
  1. Systems don’t conform 符合 to the real requirements
  2. Cost for corrections are disproportionately high
  3. Important concepts and decisions are not documented
- Waterfall model:
  A model of the software development process in which the constituent activities, typically
  1. concept phase,
  2. requirements phase,
  3. design phase,
  4. implementation phase,
  5. test phase,
  6. installation and checkout phase,

  are performed in that order, possibly with overlap but with little or no iteration

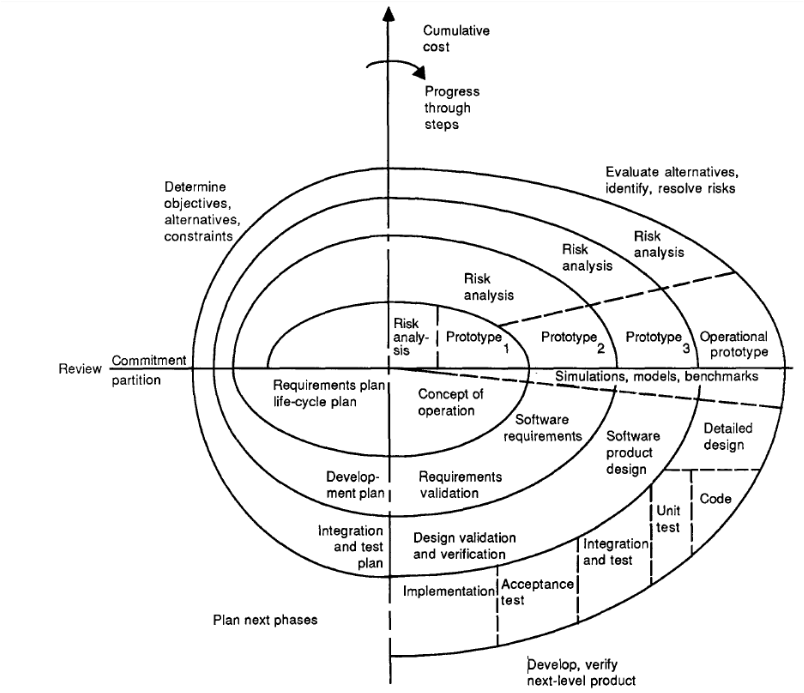


work from top left to bottom right, when errors detected, we go back to the corresponding activity
Pros:
1. Supports project management
2. Supports planning and controlling

Assumptions:

1. Requirements are known up front before design
2. Requirements rarely change
3. Users know what they want, and rarely need visualization(可运行 program 非常靠后阶段才看到)
4. The system is not so complex

- Spiral model



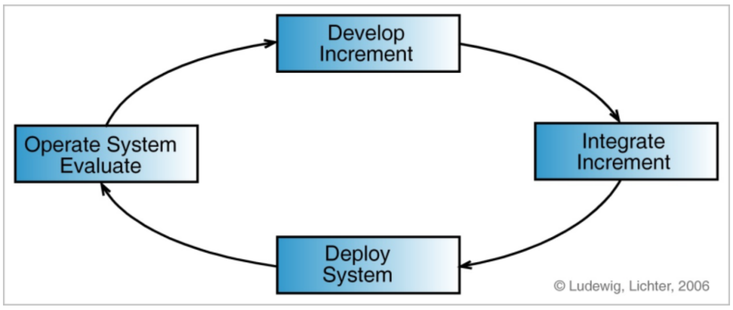
注意到其实就是 4 个 phase，逐步细化深入，但是 4 个 phase 总是在的

1. Determine objectives, alternatives, constraints
2. Evaluate alternatives, identify and resolve risks
3. Plan next phases
4. Develop and verify next-level product

Spiral model is a generic model that must be tailored
Core idea is always reacting to development risks, identify and manage as early as possible
Repeat following cycle until project finished successfully or fails:

1. Identify all risks
2. Assess all risks to get the most important one
3. Select an approach to eliminate this risk and apply this approach

- Incremental development



Characteristics and Pros:

1. Short development times
2. User feedback as soon as possible
3. Project progress shown not by documents, but by executable systems
4. Each increment implements a defined set of requirements
5. allows for software to evolve, not be produced in one huge effort
6. allows software to improve, by giving enough time to the evolutionary process itself
7. forces attention on stability, because only a stable foundation can support multiple additions
8. allows the system (a small subset of it) to run much sooner than with other processes
9. allows for the management of risks, by exposing problems earlier in the development process

- iterative development:

  iterative development is controlled continuous revision of software aiming at correcting errors early and integrate improvements

  “We get things wrong before we get things right”

  Maintenance activities are integrated in the development process

  1. recognizes the reality of changing requirements
  2. promotes early risk mitigation, by breaking down the system into mini-projects and focusing on the riskier elements first
  3. allows you to “plan a little, design a little, and code a little”
  4. encourages all participants, including testers, integrators, and technical writers to be involved earlier
  5. allows you to correct errors sooner and put into practice lessons learned in the prior iteration
  6. focuses on component architectures, not final big bang deployments

- iterative &incremental 的区别，就是一个是越画越细致，一个是先画头，再画别的部位

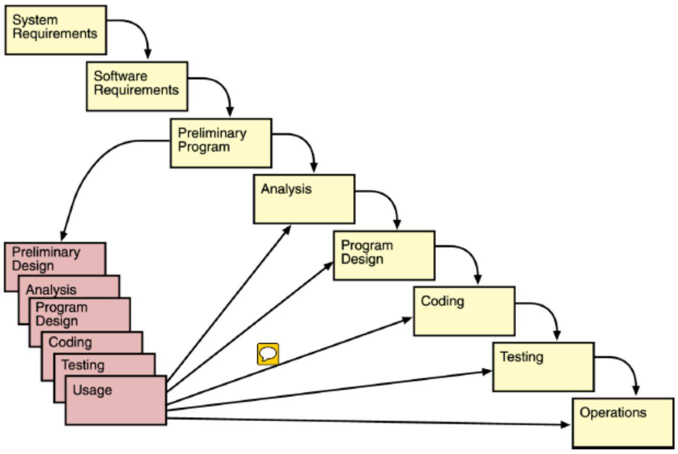
  共性是: try to integrate, validate and test as frequently as possible

  Plan to incrementally develop the system:

  1. If possible, always have a running version of the system.
  2. Integrate new functionality as soon as possible.
  3. Validate incremental versions against user requirements

- Prototyping

Do the job twice – first result provides an simulation of the final product



左右两边基本是一一对应一致的，所以就是先造一个，看会有什么问题，然后再正常认真的造

词义辨析:

prototype —A preliminary type, form, or instance of a system that serves as a model for later stages or for the final, complete version of the system.就是原型那个程序

prototyping —A hardware and software development technique in which a preliminary version of part or all the hardware or software is developed to permit user feedback, determine feasibility, or investigate timing or other issues in support of the development process.指这种开发的模式

rapid prototyping —A type of prototyping in which emphasis is placed on developing prototypes early in the development process to permit early feedback and analysis in support of the development process.快速地尽早的开发原型

idea:

1. It is easier to describe a system relatively to a similar system than to abstractly specify its requirements
2. Experiences gained in building a prototype are useful for the final system development

Implications:

1. A prototype should be executable/performable!
   A paper prototype is a very first version of a SW prototype
2. It only implements some selected attributes of the final system
3. It should be used and assessed by users/clients
4. Based on the feedback, it is modified until it fits to what the user wants.
   Then it becomes a part of the requirements specification

4 Kinds of prototypes

1. Presentation prototype
   Supports the initiation of a software project.
   Often used to convince the client that the future application system is either feasible or that its user interface and handling is in line with user requirements.
2. Functional Prototype

Is generally designed to illustrate specific aspects of the user interface or part of the functionality and helps to clarify the problem in hand.

3. Breadboard 实验板

   Models a critical technical aspect of the final system.

   Is not evaluated by the users but by the developers.

4. Pilot System 实验系统

   It is used in the application area itself as the core of the application system and is enhanced in cycles

3 Approaches to prototyping:

a. Exploratory Prototyping

   Used where the problem in hand is unclear.

   Initial ideas are used as a basis for clarifying user and management requirements with respect to the future system.就是需求和问题都还不明确

b. Experimental Prototyping

   Focuses on the technical implementation of a development goal.

   Through experimentation the users can further specify their ideas about the type of computer support required.技术实现的问题探究

c. Evolutionary Prototyping

   Not merely used as a tool in the context of a single development project.

   It is a continuous process for adapting an application system to rapidly changing organizational constraints.不断改进

- Summary

1. Software processes are an important means to manage a development project.
2. **Phases**, **roles**, **artifacts**, and **activities** are the main elements of process models.
3. Modern process models are based on **iterative** and **incremental** software development.
4. It is very important to choose the right process model.
5. The quality of a process model can influence the quality of the resulting product.
6. The process model influences project management

4 Software development processes – RUP

- The Rational Unified Process (RUP) is an iterative software development process framework
- Characteristics
  1. Defines 4 development phases and a set of disciplines.

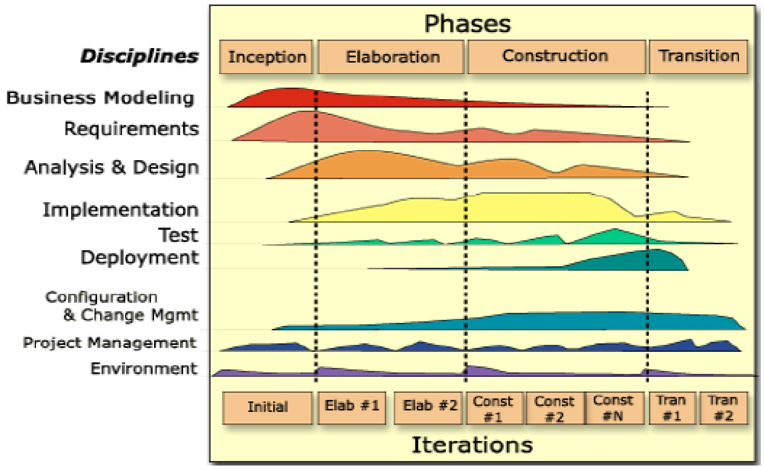
     All disciplines are performed in **each phase.**

     Each phase may have several **iterations**.
  2. Forces **iterative** and **incremental** software development
  3. Is based on **use cases** and **prototyping**

     Prototypes are used to minimize project risks and demonstrate project progress
  4. RUP assumes an object-oriented architecture and an object-oriented implementation.

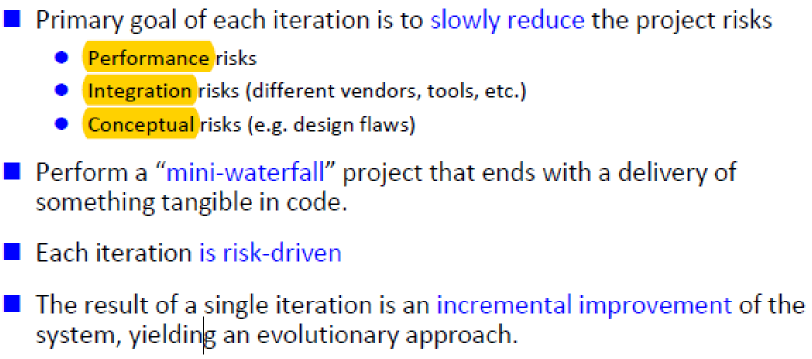
     Architecture centric

5. UML is used whenever possible

- 

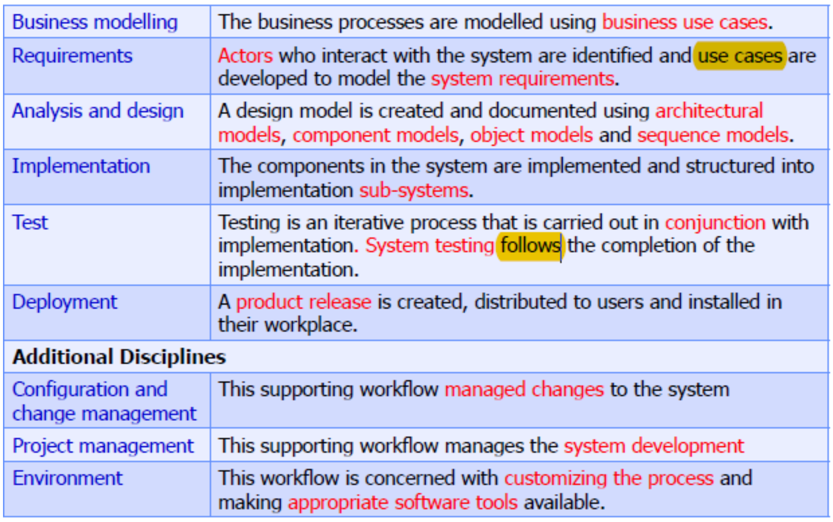
可以看出，可能有多个 iteration，而且每个阶段都包含所有 discipline，但是权重不同

- Goals and features of ***each iteration***



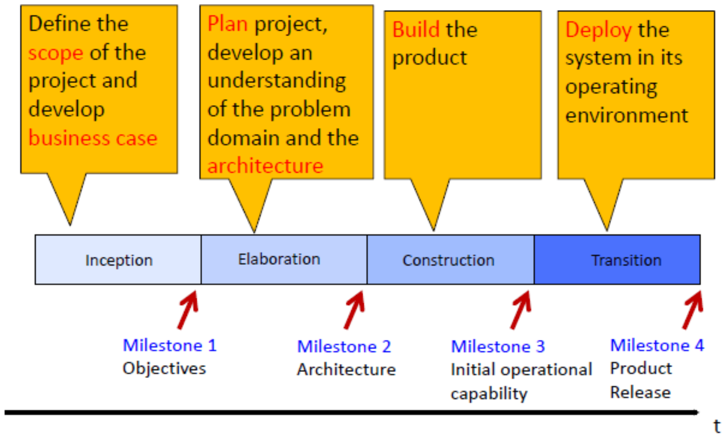
- Primary goal of each iteration is to slowly reduce the project risks
  - Performance risks
  - Integration risks (different vendors, tools, etc.)
  - Conceptual risks (e.g. design flaws)
- Perform a “mini-waterfall” project that ends with a delivery of something tangible in code.
- Each iteration is risk-driven
- The result of a single iteration is an incremental improvement of the system, yielding an evolutionary approach.

- Disciplines（下面的其实是按顺序来的）



Business modelling	The business processes are modelled using business use cases.
Requirements	Actors who interact with the system are identified and use cases are developed to model the system requirements.
Analysis and design	A design model is created and documented using architectural models, component models, object models and sequence models.
Implementation	The components in the system are implemented and structured into implementation sub-systems.
Test	Testing is an iterative process that is carried out in conjunction with implementation. System testing follows the completion of the implementation.
Deployment	A product release is created, distributed to users and installed in their workplace.
Additional Disciplines	
Configuration and change management	This supporting workflow managed changes to the system
Project management	This supporting workflow manages the system development
Environment	This workflow is concerned with customizing the process and making appropriate software tools available.

- RUP software engineering process is organized by disciplines, expressed by workflow
- RUP phases



Define the scope of the project and develop business case

Plan project, develop an understanding of the problem domain and the architecture

Build the product

Deploy the system in its operating environment

Inception | Elaboration | Construction | Transition

Milestone 1 Objectives

Milestone 2 Architecture

Milestone 3 Initial operational capability

Milestone 4 Product Release

t

按时间顺序，四个阶段

- Inception:
  - **Tasks**
    - Identify all use cases (!), work out the details of the important ones.
    - Perform a risk analysis
    - Define
      - Success criteria for the project (important deliverables)
      - Needed resources, important milestones (customer point of view)
  - **Results**
    - Project scope definition
    - Most important requirements (initial use case model)
    - Initial project plan
      - most important milestones
      - coarse cost estimation, project deliverables
    - Model of the application domain (domain model)
    - Glossary

Glossary 是术语表，感觉意思就是一些项目设定

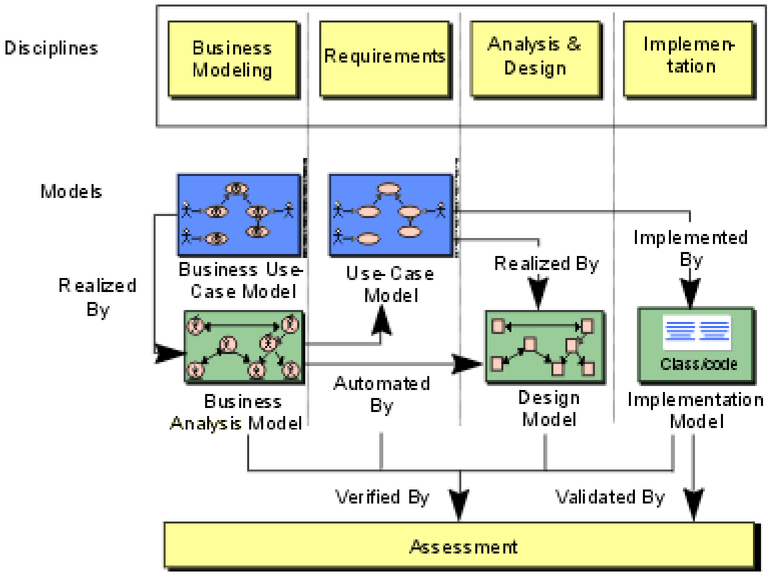
- Elaboration
  - **Goals**
    - Analysis of the application domain
    - Designing the software architecture
    - Elimination of the highest project risks
  - **Tasks**
    - Working out use cases in detail
    - Identifying and formulating non-functional requirements
    - Designing the top-level architecture
    - Construction of a first prototype
    - Creating the project plan

- Results
  - Use case analysis
    - Use Cases (80% written and reviewed by end of phase)
    - Use Case Model (80% done)
    - Non-functional requirements
  - Top-level architecture
  - Chosen development technology
  - Prototype
  - Revised Project Plan

- Construction
  - Tasks (performed iteratively)
    - Select use cases to be implemented.
    - Work out in detail the use cases (if necessary)
    - Revise the architecture (if necessary)
    - Implement the use cases
    - Test the resulting system
  - Results
    - Tested system release
    - User documentation
    - System documentation of the developed release
- Transition
  - Tasks
    - The transition phase consists of the transfer of the system to the user community
    - It includes manufacturing, shipping, installation, training, technical support and maintenance
    - Development team begins to shrink
    - Control is moved to maintenance team
    - Software updates
    - Integration with existing systems (legacy, existing versions, etc.)

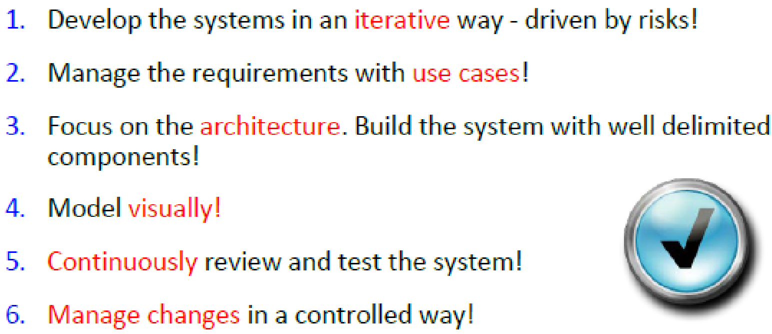
  - Results
    - Final system

- RUP structure



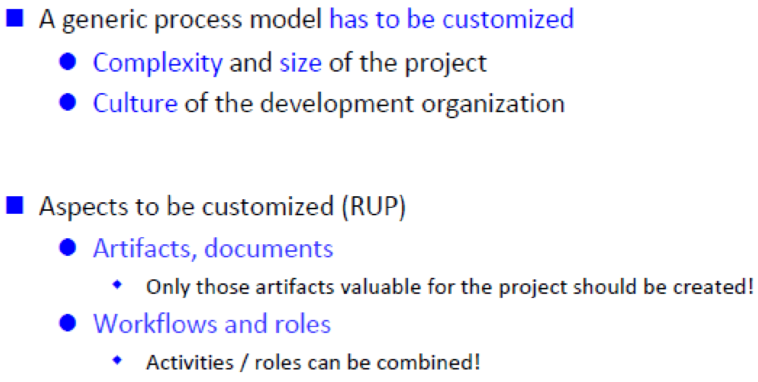
上面四个 discipline 是头四个，没有写 test 和 deployment
前面是两个 use case model，需要 verify，后面是 design 和 implementation，就是 validate

- Discipline 是如何深化指导项目的:
  比如 requirement 的流程图，其中的一部分 analyze the problem 模块进行深化
  analyze the problem 深化变成了多个 activities，并且有各自对应的输入和产出物
  然后单个 activity 比如 capture a common vocabulary 的具体描述就是份文档
- RUP's 6 best practices



1. Develop the systems in an iterative way - driven by risks!
2. Manage the requirements with use cases!
3. Focus on the architecture. Build the system with well delimited components!
4. Model visually!
5. Continuously review and test the system!
6. Manage changes in a controlled way!

- RUP customization

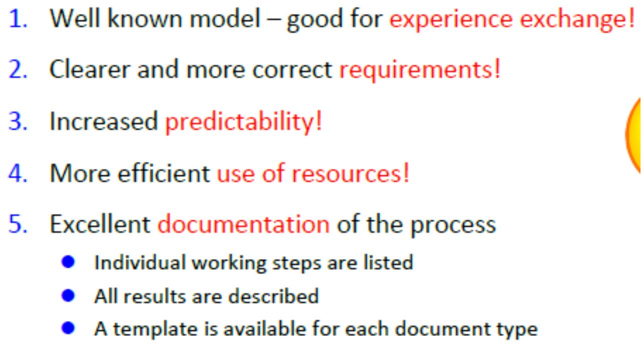


■ A generic process model has to be customized
- Complexity and size of the project
- Culture of the development organization

■ Aspects to be customized (RUP)
- Artifacts, documents
  - Only those artifacts valuable for the project should be created!
- Workflows and roles
  - Activities / roles can be combined!

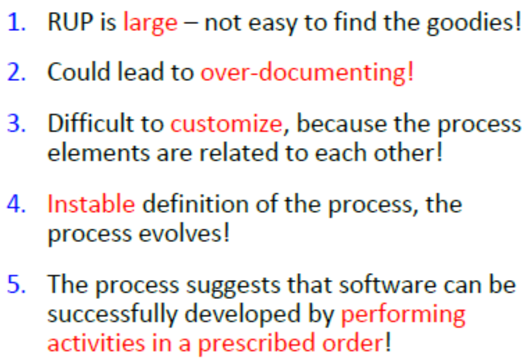
两大点，上面是根据什么来进行 customization，下面是可以改变的部分

- Pros



1. Well known model – good for experience exchange!
2. Clearer and more correct requirements!
3. Increased predictability!
4. More efficient use of resources!
5. Excellent documentation of the process
   - Individual working steps are listed
   - All results are described
   - A template is available for each document type

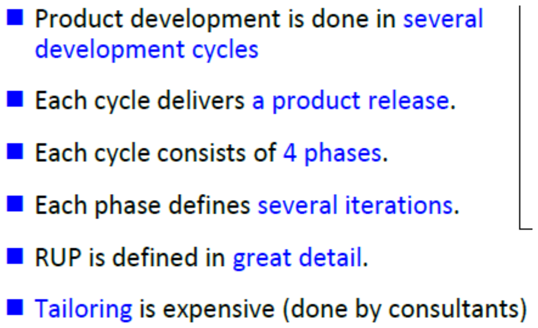
- Cons



1. RUP is large – not easy to find the goodies!
2. Could lead to over-documenting!
3. Difficult to customize, because the process elements are related to each other!
4. Instable definition of the process, the process evolves!
5. The process suggests that software can be successfully developed by performing activities in a prescribed order!

最后一条就是 process 认为项目可以在安排好的顺序下完成 activities，实际有点困难

- Summary:



- Product development is done in several development cycles
- Each cycle delivers a product release.
- Each cycle consists of 4 phases.
- Each phase defines several iterations.
- RUP is defined in great detail.
- Tailoring is expensive (done by consultants)

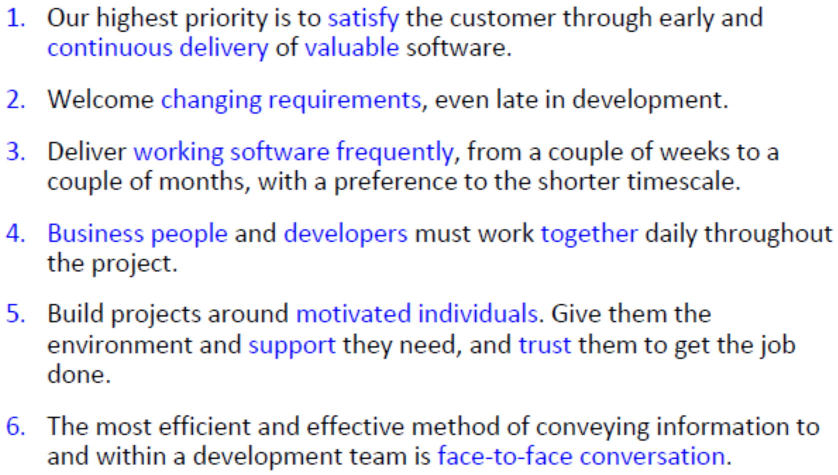
注意是多个 cycle，每一 cycle 都有 4 个 phases，每个 phase 有几个 iterations

5 Software development processes – Agile

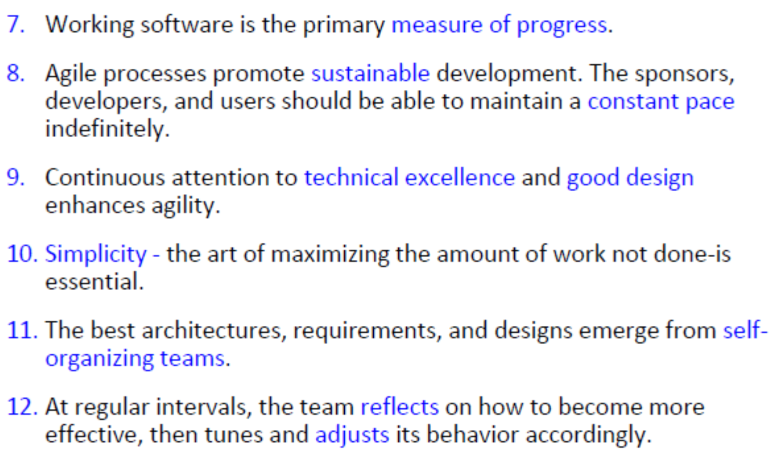
- Manifesto for agile (4 条基本，右边也有价值，但是没那么看重)

a.	Individuals and interactions	over	processes and tools
b.	Working software	over	comprehensive documentation
c.	Customer collaboration	over	contract negotiation
d.	Responding to change	over	following a plan

- Derived principles from manifesto



1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.



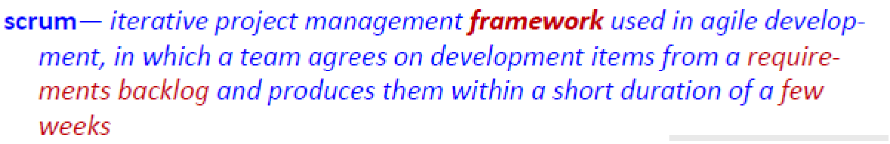
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity - the art of maximizing the amount of work not done-is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

第 8 条意思是按照一个大概的节奏来，这样可以持续

- 定义

  Agile development: software development approach based on iterative development, frequent inspection and adaptation, and incremental deliveries, in which requirements and solutions evolve through collaboration in cross-functional teams and through continuous stakeholder feedback

- Scrum

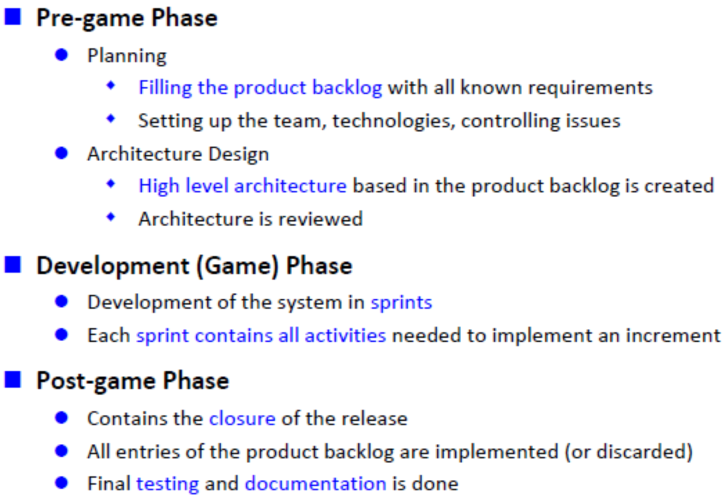


scrum— *iterative project management* ***framework*** *used in agile development, in which a team agrees on development items from a requirements backlog and produces them within a short duration of a few weeks*

Backlog: 待办的事务

Sprint: 冲刺，是 scrum 的工作单位，a time-box of one month or less during which a “Done”, useable, and potentially releasable product Increment is created

- Scrum does not define techniques and technologies, is light weight and easy to learn, but hard to apply
- Scrum phases



■ **Pre-game Phase**
- Planning
  - Filling the product backlog with all known requirements
  - Setting up the team, technologies, controlling issues
- Architecture Design
  - High level architecture based in the product backlog is created
  - Architecture is reviewed

■ **Development (Game) Phase**
- Development of the system in sprints
- Each sprint contains all activities needed to implement an increment

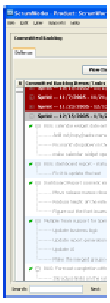
■ **Post-game Phase**
- Contains the closure of the release
- All entries of the product backlog are implemented (or discarded)
- Final testing and documentation is done

- Scrum roles

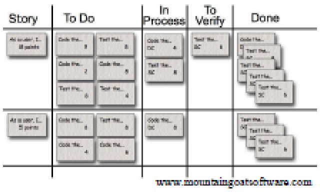
- Product Owner
    - represent all stakeholders
    - knows requirements (business and technical)
    - maintains product backlog
    - decides what to develop next (during the next sprint)
    - accepts or denies realized features
- Scrum Team
    - self-organized interdisciplinary team
    - estimates development effort for each feature
    - there is no team leader

- Scrum Master
    - ensures that the scrum process is performed as intended
    - supports the team and the product owner
    - removes blockers and problems
    - supports the team in improving its process and competencies

- Artifacts
    - Product Backlog
        - contains all product features
        - technical and non technical ones
        - estimated by e.g. story points (not person days)
    - Sprint Backlog
        - contains features to be developed in one sprint
        - determined in sprint planning meeting
        - more detailed than product backlog



    - Sprint Burndown
        - serves to monitor sprint progress
    - Release Plan
        - defines number of sprints and delivered working releases
    - Product Burndown
        - serves to monitor the whole project
- Activities
    - Sprint Planning
        - Negotiation between product owner and team about the set of features to be realized in the next sprint
        - effort is estimated by the team
    - Daily Scrum
        - 15 minutes meeting
            - What did you do since last daily scrum?
            - Are the any problems, blockers?
            - What will you do until the next daily scrum

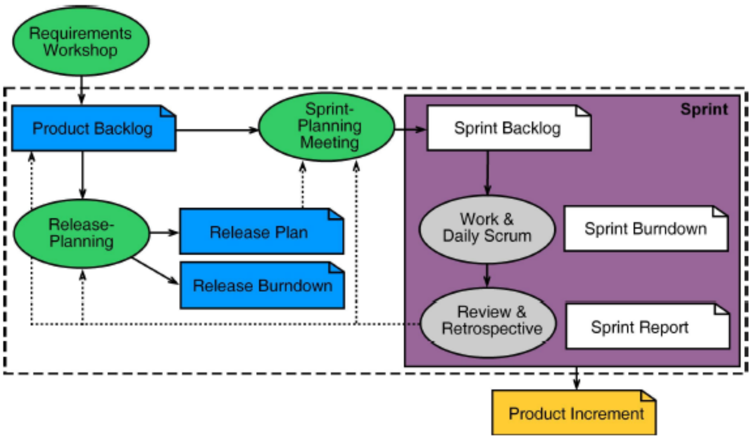


www.mountaingoatsoftware.com

    - Sprint Review
        - Presentation of the realized features to the product owner
        - Acceptance/ rejection decision for each feature of the initial sprint backlog
    - Sprint Retrospective
        - Are there improvement potentials for forthcoming sprints?

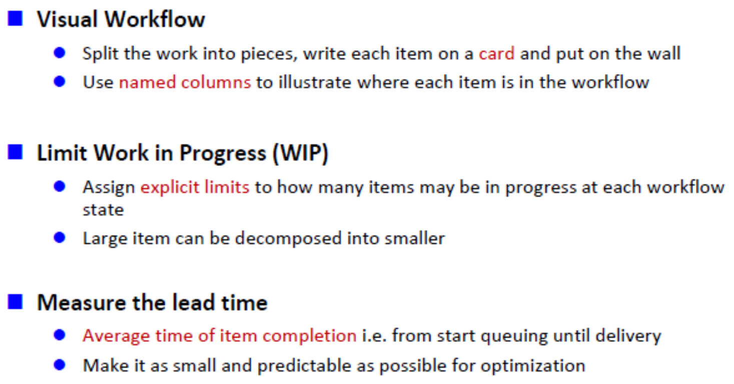
Retrospective: 回顾的

- Activities and artifacts relations



- Typical errors
  1. Session is not properly prepared
  2. Scrum master is planning (他是 process 的 support, 应该是 product owner 来)
  3. Team member dominates
  4. Too much discussion about solutions
  5. Team members only plan their own activities
  6. Product owner identified activities (应该是 scrum team 来)
  7. Product owner not available
  8. Activities are too vague or too big
- Summary:
  - **Scrum is applied in industry with success!**
  - **Scrum is not applicable for all kinds of projects!**
    - size of project, complexity of teams, distributed development
    - team size of each single scrum team should not exceed 10 persons
  - **Scrum of scrums is used to manage bigger projects**
    - a team of product owners
    - a team of scrum masters
  - **Essential requirements**
    - product owner has to be available
      - at least for the planning meetings and the scrum review meetings
    - team members need both high technical and social skills
    - experienced scrum master

  单个 scrum team 不能超过 10 人，大的项目可以用 scrum of scrums, 那时 product owners and scrum masters 就都是一团队的人
- Kanban



- **Visual Workflow**
  - Split the work into pieces, write each item on a card and put on the wall
  - Use named columns to illustrate where each item is in the workflow
- **Limit Work in Progress (WIP)**
  - Assign explicit limits to how many items may be in progress at each workflow state
  - Large item can be decomposed into smaller
- **Measure the lead time**
  - Average time of item completion i.e. from start queuing until delivery
  - Make it as small and predictable as possible for optimization

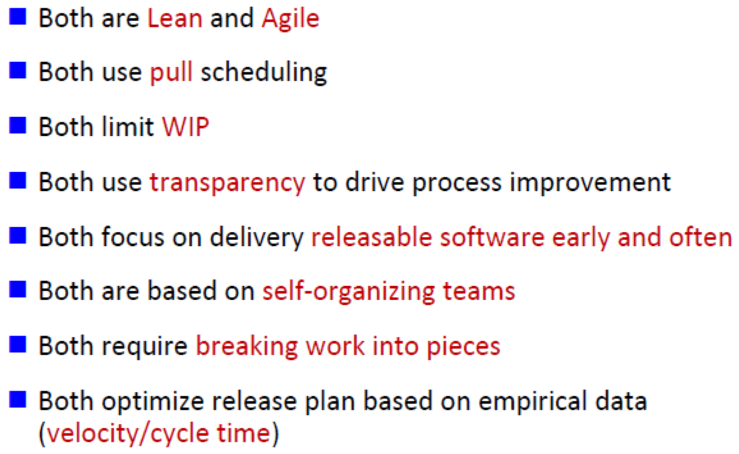
三个 kanban 的特点：
Work 化成 item, i.e. feature 放在不同列里表示 workflow
限制每个 column， 也就是 workflow state 的 item 数量（各列间可以不同）
看的是 item 的平均 queue 到 delivery 间的时长

- When a feature/user story/item is large:

  1. Takes longer than a couple of days to complete
  2. Requires that multiple people collaborate on its completion

  Then decompose that step into cards(i.e. small tasks) to track independently 比如不同的 card 放在 queue, in progress, complete 等列下面

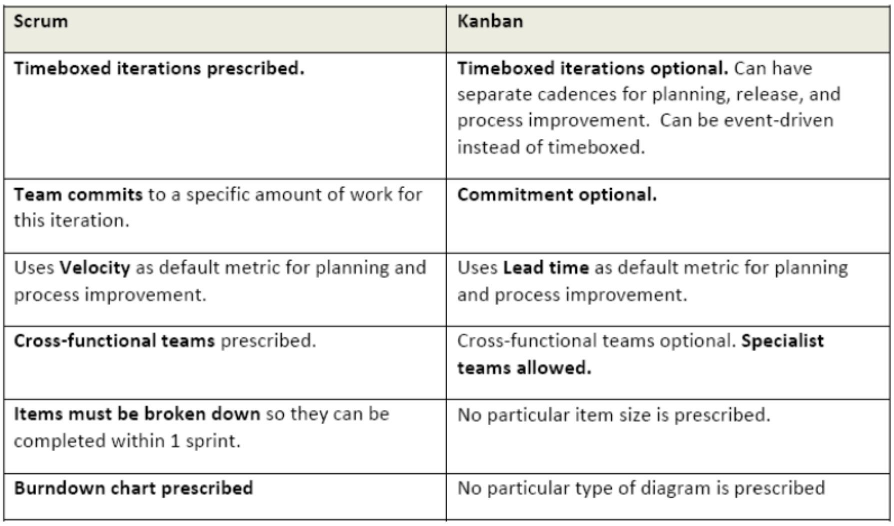
- Similarities of Scrum and Kanban



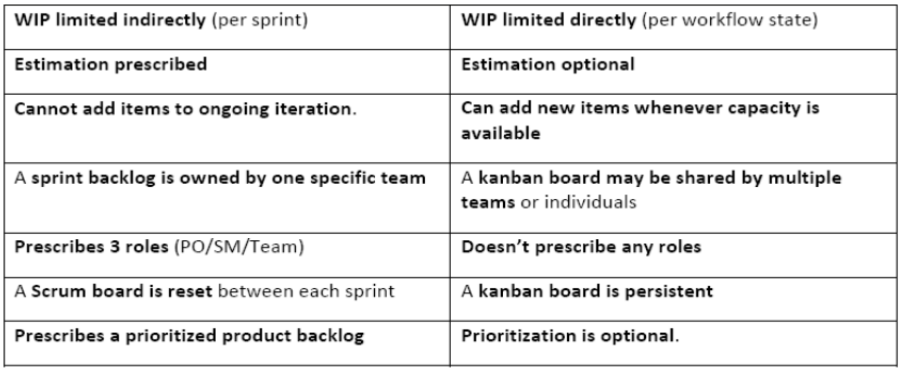
- Both are Lean and Agile
- Both use pull scheduling
- Both limit WIP
- Both use transparency to drive process improvement
- Both focus on delivery releasable software early and often
- Both are based on self-organizing teams
- Both require breaking work into pieces
- Both optimize release plan based on empirical data (velocity/cycle time)

Lean: 瘦的
Pull 就是说是，因为当前 column 有 wip 限制，所以要等自己判断有空了才拉去 item
Empirical: 经验主义的， 也就是根据过去的数据， 分别是速度/时间判断之后的时间

- Differences between Scrum and Kanban



Scrum	Kanban
Timeboxed iterations prescribed.	**Timeboxed iterations optional.** Can have separate cadences for planning, release, and process improvement. Can be event-driven instead of timeboxed.
Team commits to a specific amount of work for this iteration.	**Commitment optional.**
Uses **Velocity** as default metric for planning and process improvement.	Uses **Lead time** as default metric for planning and process improvement.
Cross-functional teams prescribed.	Cross-functional teams optional. **Specialist teams allowed.**
Items must be broken down so they can be completed within 1 sprint.	No particular item size is prescribed.
Burndown chart prescribed	No particular type of diagram is prescribed



WIP limited indirectly (per sprint)	**WIP limited directly** (per workflow state)
Estimation prescribed	**Estimation optional**
Cannot add items to ongoing iteration.	**Can add new items whenever capacity is available**
A **sprint backlog is owned by one specific team**	A **kanban board may be shared by multiple teams** or individuals
Prescribes 3 roles (PO/SM/Team)	**Doesn't prescribe any roles**
A **Scrum board is reset** between each sprint	A **kanban board is persistent**
Prescribes a prioritized product backlog	**Prioritization is optional.**

Cadence：节奏，第一条就是 scrum 是每个 sprint 有大致时间，但 kanban 是不同列下面的 item 不同时长

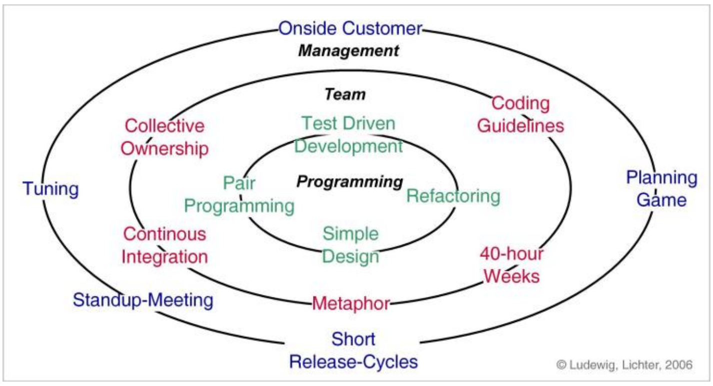
- Extreme programming (XP)

  - **Exploration Phase**
    - Customers write stories to be included in the 1. release
    - Development team familiarizes with the technology, tools, platforms etc.
  - **Planning Phase**
    - Scope of 1. small release is defined.
    - Developers estimate the effort and duration to implement the stories (< 2 month)
  - **Iterations to Release Phase**
    - Includes all iterations before the first release
    - First iteration creates a system with “complete” architecture
    - Based on the prioritization the stories are added
    - Functional tests created by the customer are run at the end of every iteration.

- **Productionizing Phase**
  - Extra testing is performed before releasing the system to the customer
  - New changes may be included in the release or not
  - Postponed ideas and changes are documented for later implementation
- **Maintenance Phase**
  - System must be kept running and new iterations are performed in parallel.
  - Often the development team changes.

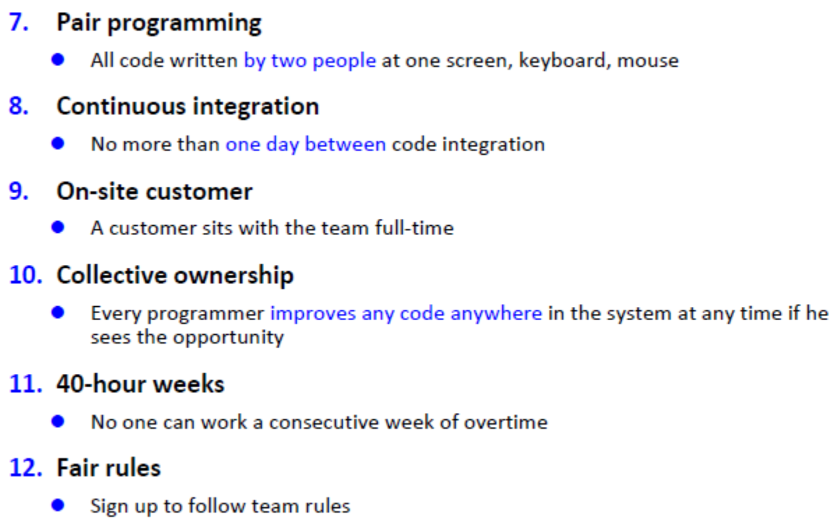
最后可以加一个 death phase，也就是 final release，不再维护

XP practices 14 个指导方针



© Ludewig, Lichter, 2006

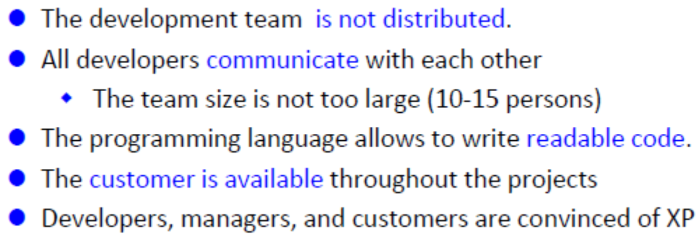
1. **Planning game**
   - Customers decide the scope and timing of releases based on estimates made by programmers
   - Programmers implement only functionality demanded by stories in this iteration
2. **Small releases**
   - Working system early
   - Releases anywhere from daily to monthly
3. **Metaphor**
   - A story that customers, programmers, and managers can tell about how the system works
   - System shape defined by a metaphor shared by the customer and programmers
4. **Simple design**
   - Communicates everything the programmers want to communicate
   - Contains no duplicate code
   - Has the fewest possible classes and methods
   - Say everything once and only once
5. **Test driven development**
   - Programmers write unit tests
   - Customers write functional tests
6. **Refactoring**
   - System evolves through transformations of existing designs
   - A change that leaves system behavior unchanged, but enhances simplicity, flexibility, understandability, and/or performance



7. **Pair programming**
   - All code written by two people at one screen, keyboard, mouse
8. **Continuous integration**
   - No more than one day between code integration
9. **On-site customer**
   - A customer sits with the team full-time
10. **Collective ownership**
    - Every programmer improves any code anywhere in the system at any time if he sees the opportunity
11. **40-hour weeks**
    - No one can work a consecutive week of overtime
12. **Fair rules**
    - Sign up to follow team rules

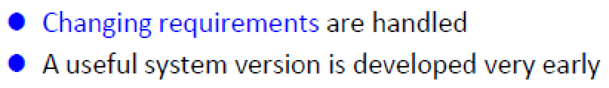
上面只有 12 条，没有 tuning and standup-meeting
Fair rules i.e. coding guidelines, onside customer i.e. on-site customer, small releases i.e. short release-cycle
Metaphor 指 customers, programmers, managers 三方共同的关于系统的 shape 的 story

Prerequisites:



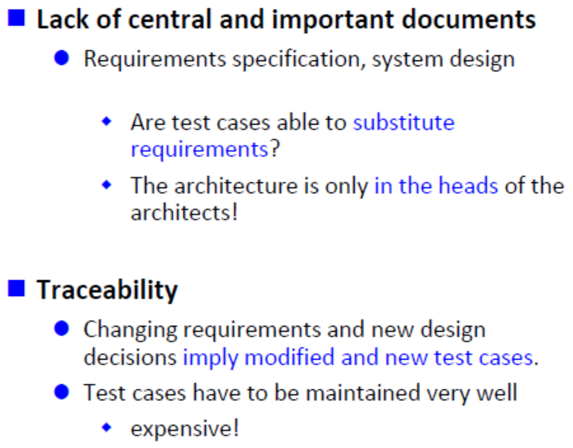
- The development team is not distributed.
- All developers communicate with each other
  - The team size is not too large (10-15 persons)
- The programming language allows to write readable code.
- The customer is available throughout the projects
- Developers, managers, and customers are convinced of XP

Pros:



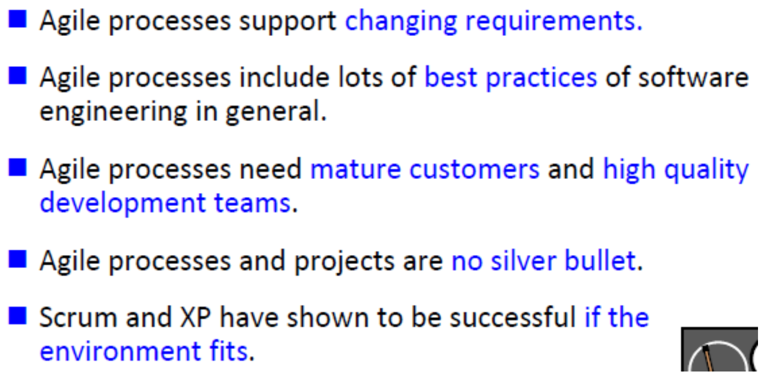
- Changing requirements are handled
- A useful system version is developed very early

Cons:



- **Lack of central and important documents**
  - Requirements specification, system design
    - Are test cases able to substitute requirements?
    - The architecture is only in the heads of the architects!
- **Traceability**
  - Changing requirements and new design decisions imply modified and new test cases.
  - Test cases have to be maintained very well
    - expensive!

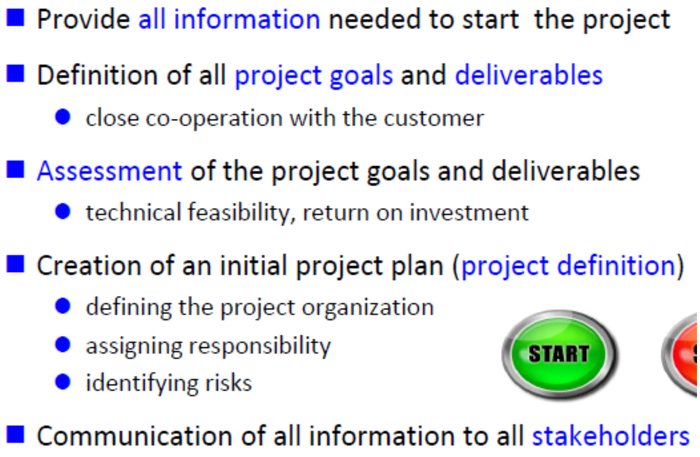
- Summary of Agile



- Agile processes support changing requirements.
- Agile processes include lots of best practices of software engineering in general.
- Agile processes need mature customers and high quality development teams.
- Agile processes and projects are no silver bullet.
- Scrum and XP have shown to be successful if the environment fits.

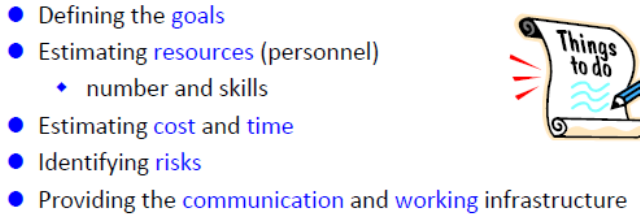
6 Project preparation and initialization

- The preparation and initialization phase is not part of the project
- Objectives of preparation phase



- Provide all information needed to start the project
- Definition of all project goals and deliverables
  - close co-operation with the customer
- Assessment of the project goals and deliverables
  - technical feasibility, return on investment
- Creation of an initial project plan (project definition)
  - defining the project organization
  - assigning responsibility
  - identifying risks
- Communication of all information to all stakeholders

- Tasks of preparation phase



- Defining the goals
- Estimating resources (personnel)
  - number and skills
- Estimating cost and time
- Identifying risks
- Providing the communication and working infrastructure

Things to do

如果 budget and deadlines 已知，则必须反向计划（而不是看目标，然后估计时间和花费）

- Project definition including 7 parts:
  1. Project definition(简述目标和产出)
  2. Completion criteria
  3. Goals and objectives
  4. Budget and schedule
  5. Roles and responsibilities
  6. Constraints, risks and dependencies
  7. Scope and deliverables
- Goal: a project goal is everything the project has to produce and to deliver
  要求： all stakeholders be involved, needs of each group can vary, goals can be

contradictory, contradictions need to be detected and resolved, pm make sure that all goals are achievable

- Goal should be precise, documented in the project definition. Non-goals help to make goals more accurate, e.g. we do not maintain the components we developed.
- SMART properties:
    1. Specific
    2. Measurable
    3. Attainable 可实现的
    4. Relevant 在已有条件下现实的
    5. Timed 时间上可达到的
- Goals vs objective

  Similarity:
    1. Be SMART
    2. Well-worded
    3. More of art than science

  Difference:

  Goal: abstract, high-level statements describe overall context of project, aligned with business goals.

  Objective: fine-grained, lower-level statements describe specific, tangible products and deliverables
- Deliverables

  Either tangible or intangible

  Must be achievable
- Scope

  What is and what is not included in project

  Describe the boundaries that all stakeholders agree, tightly aligned to the goals and objectives
- Assumptions

  Are factors that, for planning purposes, are considered to be true, real, or certain.

  Characteristics:
    1. They affect all aspects of planning
    2. are part of the progressive elaboration of the project
    3. frequently identified, documented, and validated as part of planning process
    4. involve a degree of risk

  e.g. working time of team, availability of work station
- Constraints

  Are factors that limit the project team's options

  e.g. cost, time, human resource, technical, dependencies on other projects
- Completion criteria

  Deliverables that are defined by customer as crucial

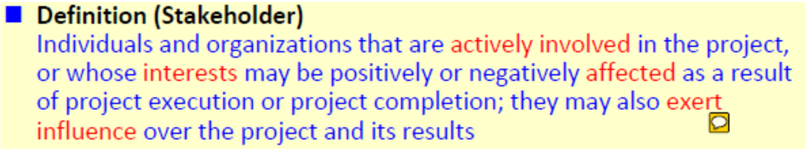
  It is a communication tool between stakeholders – agree on when a particular deliverable is done

  It can be set on explicitly on different levels: track, major phases, customer acceptance,

entire projects etc.
e.g. all unit tests are passed, no major bug while running

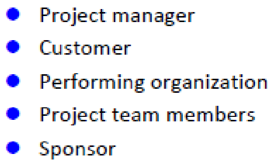
- Project definition is an overview document that sets the scope of the project
- Good project definition - Checklist:
    1. Goal SMART?
    2. Non-goals defined?
    3. Contradictions resolved?
    4. Schedule and budget estimates realistic?
    5. Stakeholders accept all goals and estimates?
    6. Important risks identified and assessed?
- Summary:
    1. Preparation phase is essential
    2. Defining goals (or assumptions, constraints) is important
    3. Results of preparation phase are: **project definition, initial project plan**
    4. Preparation phase consumes resources

7 Stakeholder management

- 

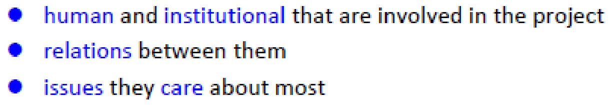
  ■ **Definition (Stakeholder)**
  Individuals and organizations that are actively involved in the project, or whose interests may be positively or negatively affected as a result of project execution or project completion; they may also exert influence over the project and its results

  上面的定义说明了，除了主动参与的，那些会被项目施加影响的人或组织也都是 stakeholder
- Key stakeholders

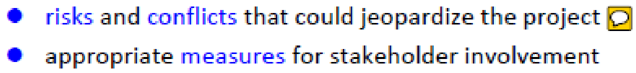
  - Project manager
  - Customer
  - Performing organization
  - Project team members
  - Sponsor
- Stakeholder analysis 的目标或者说意义

  Identify:

  - human and institutional that are involved in the project
  - relations between them
  - issues they care about most

  Help identify:

  - risks and conflicts that could jeopardize the project
  - appropriate measures for stakeholder involvement

  It is used as input for the communication plan.
- Approach of Stakeholder analysis – 3 个方面，也就是判断 interest，power 以及对待的 measure

  1. Identify interest

- stakeholders and their interests on the project
- project's benefit(s) to the stakeholder
- project activities that might cause damage or conflict for the stakeholder

2. Assess power
    - Identify stakeholder ‘s role in the project
    - Evaluate stakeholder ‘s influence in the project
    - Evaluate the likelihood and impact of stakeholder’s negative response to the project
3. Identify measures to involve them
    - Provide information
    - Motivate
    - Monitor
    - Work together
    - ...

- Techniques:
- Socio-diagram

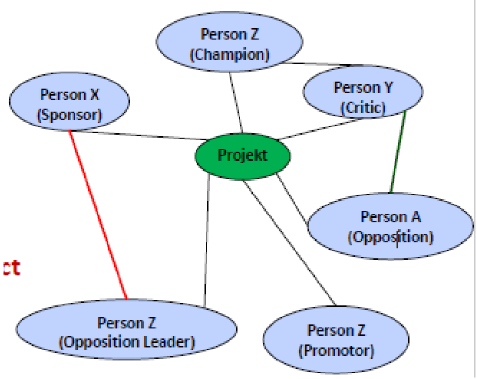
Map to 6 roles:

- Champion
- Promoter
- Sponsor
- Critic
- Opposition
- Opposition Leader

Champion 意思是拥护者，领导者

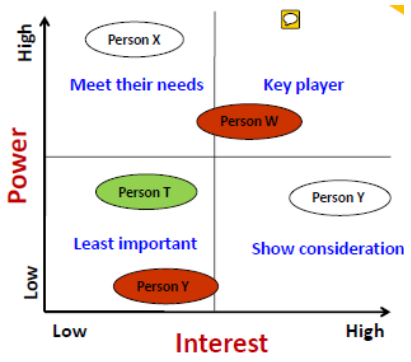
Distance to the project is the degree of involvement

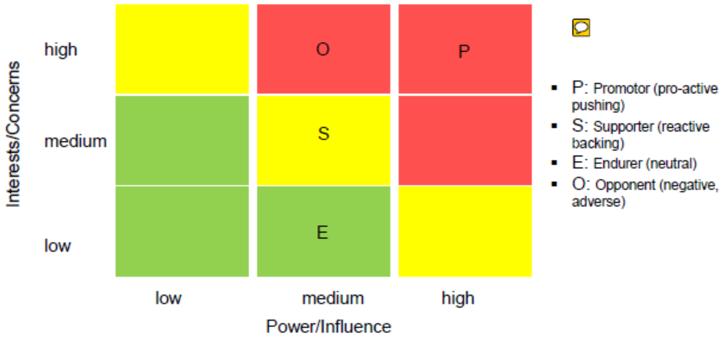
Relation between stakeholders: 1. red-negative; 2. green-positive; 3. black-neutral



- Power matrix

Power, interest, attitude.



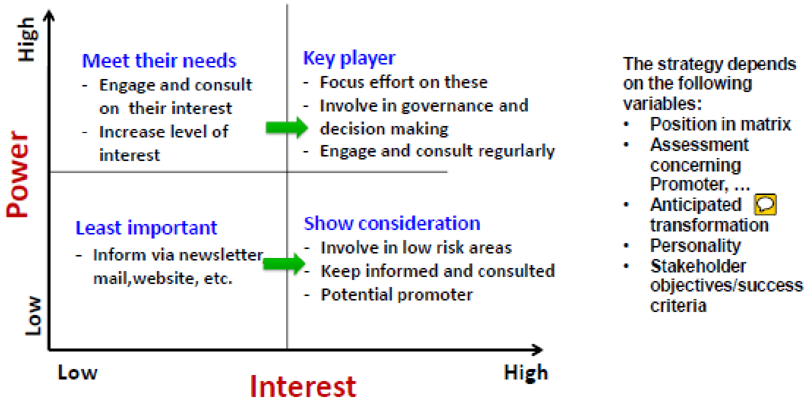


Interests/Concerns	low	medium	high
high	🟨	🟥 O	🟥 P
medium	🟩	🟨 S	🟥
low	🟩	🟩 E	🟨

Power/Influence

- P: Promotor (pro-active pushing)
- S: Supporter (reactive backing)
- E: Endurer (neutral)
- O: Opponent (negative, adverse)

- 根 据 interest， power， objective， personality， expected reaction, anticipated transformation(预期转变)来安排不同的 measure，如下图中黑色小字



Meet their needs
- Engage and consult on their interest
- Increase level of interest

Key player
- Focus effort on these
- Involve in governance and decision making
- Engage and consult regurlarly

Least important
- Inform via newsletter, mail,website, etc.

Show consideration
- Involve in low risk areas
- Keep informed and consulted
- Potential promoter

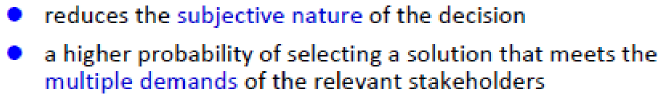
The strategy depends on the following variables:
- Position in matrix
- Assessment concerning Promoter, ...
- Anticipated transformation
- Personality
- Stakeholder objectives/success criteria

- Summary:
  Stakeholder analysis can identify roots for risks very early
  We need to consider multiple demands of the stakeholders

8 Decision analysis

- Decision analysis is a formulization of common sense for decision problems which are too complex for informal use of common sense
- 特点
  - focuses on an analysis of possible decision (alternatives)
  - uses a formal evaluation process
  - evaluates possible decisions against established criteria

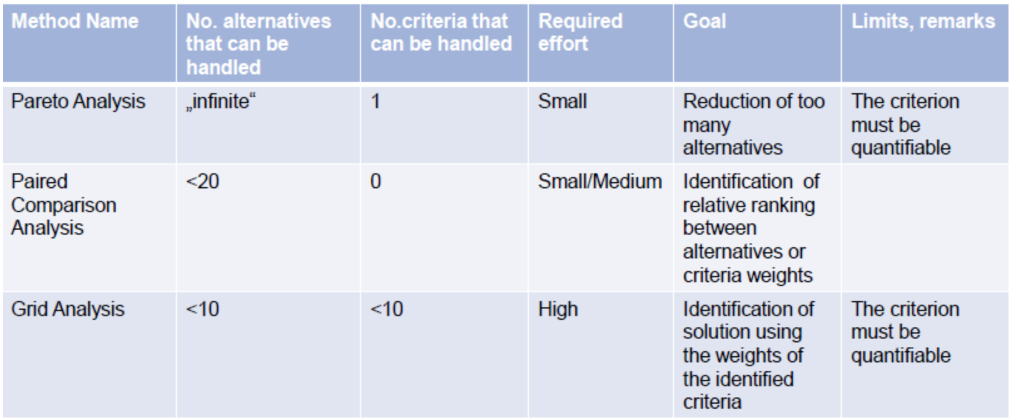
- Reasons, Pros, benefits



- reduces the subjective nature of the decision
- a higher probability of selecting a solution that meets the multiple demands of the relevant stakeholders

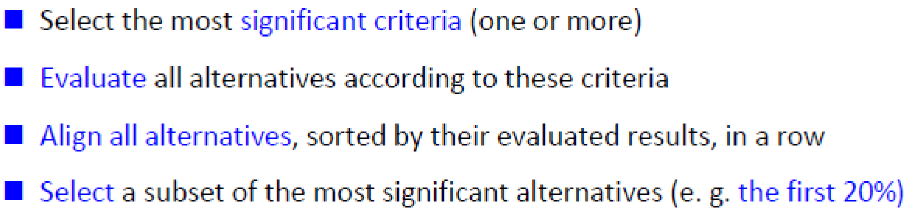
- Process

Start	
Plan the decision	1. Describe the decision issue 2. identify stakeholders that participate in the decision process
Identify alternative solutions	1. stakeholder involvement 2. literature searching 3. brainstorming sessions
Establish evaluation criteria	1. Identify a set of criteria for evaluation 2. Rank them to reflect their importance (1..10, low,high or -,0,+)
Select evaluation methods	Select best fit evaluation method (3 methods)
Identify solution	Evaluate identified alternatives (criteria and method) Select the best solution Analyze the selected solution for side effects (plus/minus side effect and risk analysis)side effect 不能是已经存在与 criteria 中的
End	



Method Name	No. alternatives that can be handled	No.criteria that can be handled	Required effort	Goal	Limits, remarks
Pareto Analysis	„infinite“	1	Small	Reduction of too many alternatives	The criterion must be quantifiable
Paired Comparison Analysis	<20	0	Small/Medium	Identification of relative ranking between alternatives or criteria weights	
Grid Analysis	<10	<10	High	Identification of solution using the weights of the identified criteria	The criterion must be quantifiable

- Pareto analysis 减少过多的选择

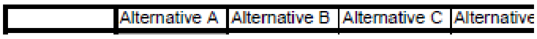


- Select the most significant criteria (one or more)
- Evaluate all alternatives according to these criteria
- Align all alternatives, sorted by their evaluated results, in a row
- Select a subset of the most significant alternatives (e. g. the first 20%)

对比的表中写的是 1 个 criteria，但是这里也说明可以是多个，只要能很好排序就好了

- Paired comparison analysis 从多个中挑好的或者用于处理 criteria 以给别的 method 进一步用

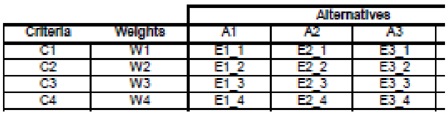
- Draw a table of alternatives
- Compare alternatives in the rows with the alternatives in the first column
- Decide which is the most important and identify the difference in importance from 0 (no difference) to 3 (major difference)
- Add up the total of the scores for each alternative (normalize values if appropriate)



	Alternative A	Alternative B	Alternative C	Alternative

说白了就是画个大表格，两两比较，看谁胜出，以及胜利程度，可以平手 0

- Grid analysis 通过已经有权重的 criteria 挑选 alternatives
  - Collect all relevant alternatives and selection criteria
  - Evaluate each alternative for each criterion and fill in the table with the appropriate evaluation
  - Calculate the sum products of the contributions of each alternative (e.g. A1 = SP_A1 = W1*E1_1 + W2*E1_2 + ...+ W5*E1_5)
  - The alternative with the highest overall sum product is the best candidate solution



		Alternatives		
Criteria	Weights	A1	A2	A3
C1	W1	E1_1	E2_1	E3_1
C2	W2	E1_2	E2_2	E3_2
C3	W3	E1_3	E2_3	E3_3
C4	W4	E1_4	E2_4	E3_4

就是行是 criteria 带 weight，列是 alternatives，计算乘积的和，然后比较

- Summary:

和上一章相同

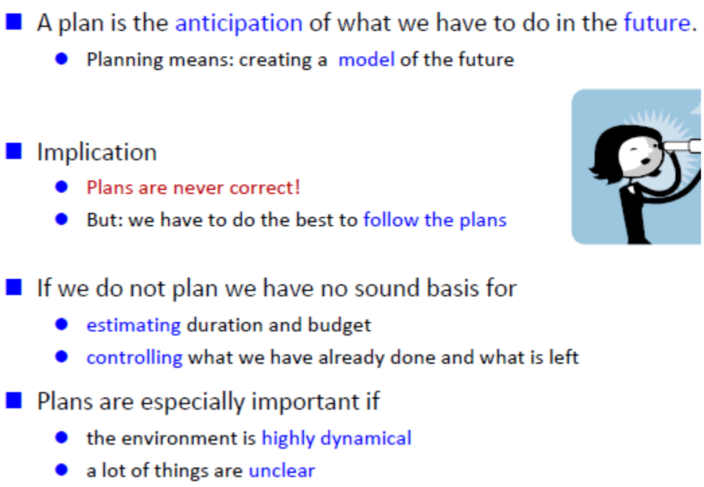
Identify roots for risks very early

Systematic approach to take decisions

Consider multiple demand of stakeholders

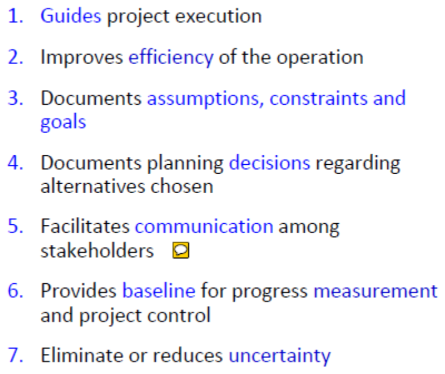
9 Planning foundation

- Project planning – characteristics
  - Continuous activity from initial concept to system delivery.
    - Plans must be regularly revised as new information becomes available.
  - Various different types of plan may be developed
    - Completing the main software project plan that is concerned with schedule and budget.
  - Probably the most time-consuming project management activity.



- A plan is the anticipation of what we have to do in the future.
  - Planning means: creating a model of the future
- Implication
  - Plans are never correct!
  - But: we have to do the best to follow the plans
- If we do not plan we have no sound basis for
  - estimating duration and budget
  - controlling what we have already done and what is left
- Plans are especially important if
  - the environment is highly dynamical
  - a lot of things are unclear

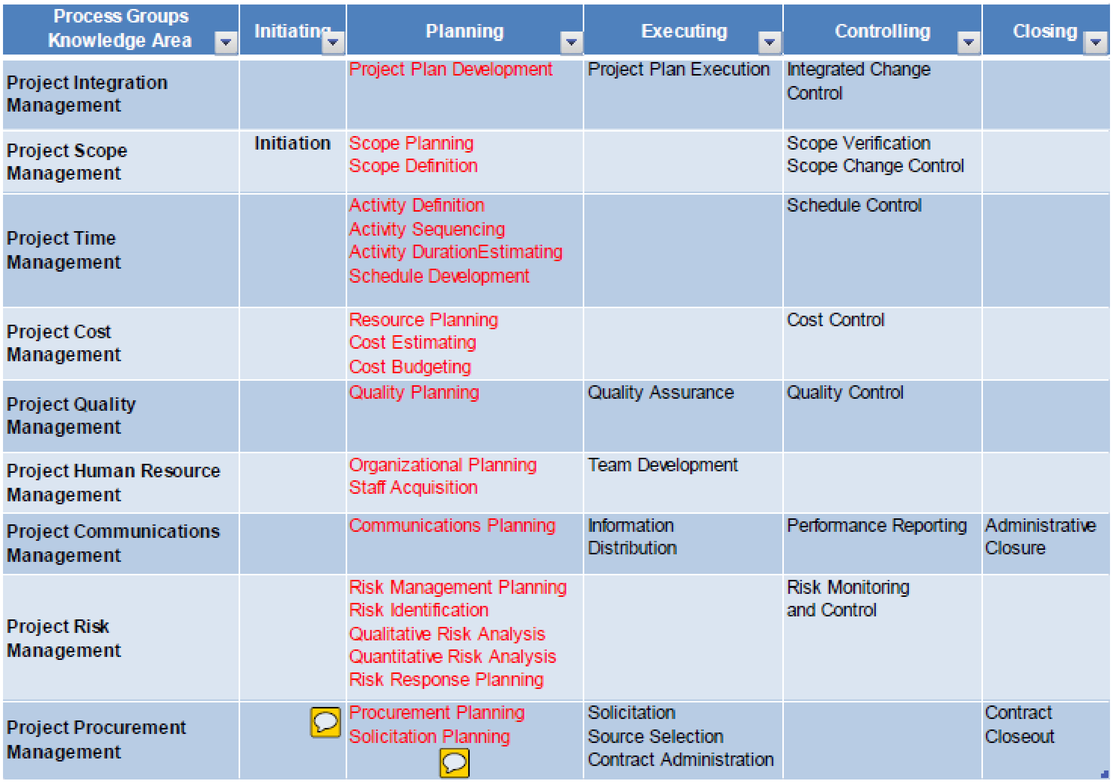
- Issues to be planned (what to do, what to do first?)
  1. Deliverables
  2. Dates
  3. Resources
- Project planning – purposes



1. Guides project execution
2. Improves efficiency of the operation
3. Documents assumptions, constraints and goals
4. Documents planning decisions regarding alternatives chosen
5. Facilitates communication among stakeholders
6. Provides baseline for progress measurement and project control
7. Eliminate or reduces uncertainty

Facilitate: 促进

- Aspects of project planning (说白了就是planning涉及到方方面面，各种的management都有 planning 的一部分)



Process Groups Knowledge Area	Initiating	Planning	Executing	Controlling	Closing
Project Integration Management		Project Plan Development	Project Plan Execution	Integrated Change Control	
Project Scope Management	Initiation	Scope Planning Scope Definition		Scope Verification Scope Change Control	
Project Time Management		Activity Definition Activity Sequencing Activity DurationEstimating Schedule Development		Schedule Control	
Project Cost Management		Resource Planning Cost Estimating Cost Budgeting		Cost Control	
Project Quality Management		Quality Planning	Quality Assurance	Quality Control	
Project Human Resource Management		Organizational Planning Staff Acquisition	Team Development		
Project Communications Management		Communications Planning	Information Distribution	Performance Reporting	Administrative Closure
Project Risk Management		Risk Management Planning Risk Identification Qualitative Risk Analysis Quantitative Risk Analysis Risk Response Planning		Risk Monitoring and Control	
Project Procurement Management		Procurement Planning Solicitation Planning	Solicitation Source Selection Contract Administration		Contract Closeout

- Scope, time, cost 等的 planning 有自己的小 iteration，而组成的 project planning 有一个大的 iteration
- Project plan

■ Definition (Project Plan)
A document that describes the technical and management approach to be followed for a project.

■ The plan typically defines
- the work to be done,
- the resources required,
- the methods to be used,
- the procedures to be followed,
- the schedules to be met,
- and the way that the project will be organized.

The project plan is central and important!
- It's the basis to measure the progress
- It's formal and approved

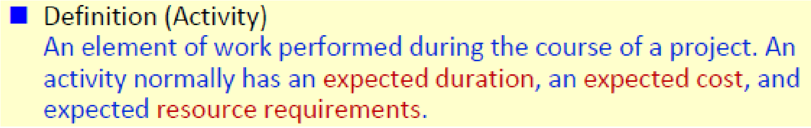
- Baseline

■ Definition (Baseline)
The originally approved plan plus or minus approved changes.

■ Baselines are used
- to compare the actual performance
- to give forecasts of the project with the original plan

baseline 就是随着时间变化的修正过的 project plan

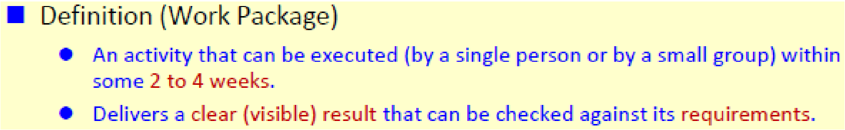
- 3 planning concepts
  1. Activities:



■ Definition (Activity)
An element of work performed during the course of a project. An activity normally has an expected duration, an expected cost, and expected resource requirements.

Activities 不断 decomposed into sub-activities，最后变成 work package 就停了。

2. Work package:



■ Definition (Work Package)

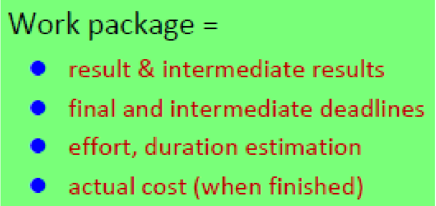
- An activity that can be executed (by a single person or by a small group) within some 2 to 4 weeks.
- Delivers a clear (visible) result that can be checked against its requirements.

Work package

a. has well-defined inputs and outputs

b. contract between **project manager** and **developer**

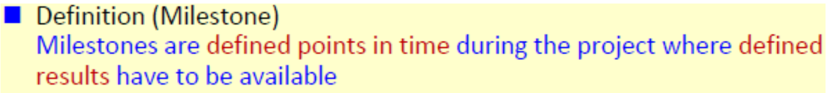
c. basis for estimation of remaining effort, controlling, and workload balancing



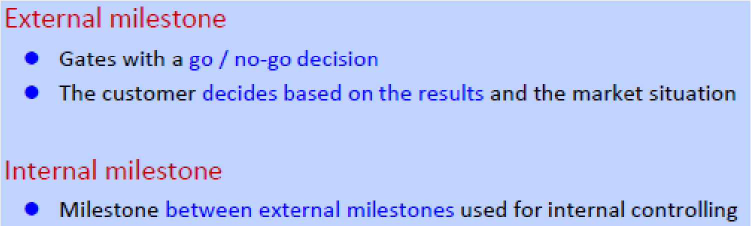
Work package =

- result & intermediate results
- final and intermediate deadlines
- effort, duration estimation
- actual cost (when finished)

3. Milestone:



■ Definition (Milestone)
Milestones are defined points in time during the project where defined results have to be available



External milestone

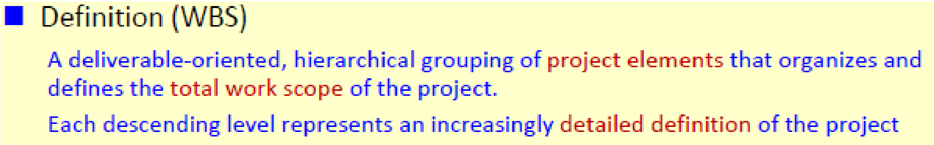
- Gates with a go / no-go decision
- The customer decides based on the results and the market situation

Internal milestone

- Milestone between external milestones used for internal controlling

外部 milestone 是顾客定的，决定进不进入下一阶段。内部 milestone 自己定用来 control，是外部的细分。

- Decomposition of deliverables:
  1. Identify major deliverables
  2. Check if adequate for cost and time estimation
  3. Yes, then validate decomposition (可以看成复查)
  4. No, identify constituent components and go to 2
- WBS (work breakdown structure)



■ Definition (WBS)

A deliverable-oriented, hierarchical grouping of project elements that organizes and defines the total work scope of the project.

Each descending level represents an increasingly detailed definition of the project

可以有两种划分方式，但它们都会有个大的 project management 这个 activity

1. Product components
2. Process phases

- WBS 和别的 activity 的关系

  WBS 需要的 basis

  1. Make-or-buy decisions
  2. System solution

  WBS 相互 consistent 的

  1. Project organization

  WBS 互为 basis 的

  1) Change management
  2) Plans, reviews

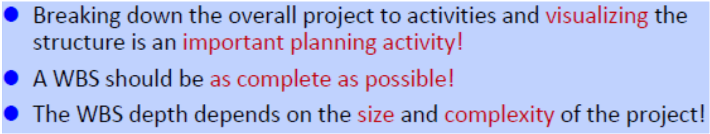
  WBS 是其 basis 的

  1) Status reporting
  2) Estimations
  3) Schedule

- WBS 2 rule to validate

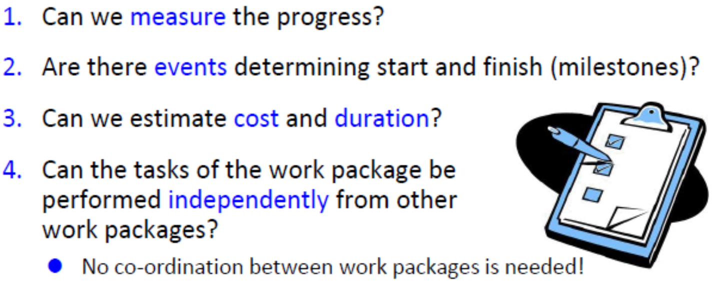
  Disjunction rule

  Completion rule



- Breaking down the overall project to activities and visualizing the structure is an important planning activity!
- A WBS should be as complete as possible!
- The WBS depth depends on the size and complexity of the project!

- Criteria to check WBS work packages



1. Can we measure the progress?
2. Are there events determining start and finish (milestones)?
3. Can we estimate cost and duration?
4. Can the tasks of the work package be performed independently from other work packages?
   - No co-ordination between work packages is needed!

  不过上面的不可能对所有 work packages 都成立

- Advices and good practices on WBS

  1) Stop decomposing if it can be easily planned
  2) Decompose work packages in tasks
  3) Check WBS against the project definition
  4) Develop WBS in the team jointly

- Summary:

  Plans are never correct, but we plan and try to follow.

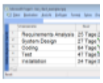
  Plan contains all information to perform a project, and it need updated regularly

  WBS is an important planning means

10 Precedence planning & scheduling

- Precedence planning: logical sequence of work packages
  Scheduling: temporal sequence of work packages
- Dependencies 成因
  1) Technological
  2) Organizational
  3) Mandatory (hard logic)
  4) Discretionary 自由决定的 (soft logic)
- 3 Activity planning techniques
  - Milestone chart
    - simple chart of most important events
    - Purpose: **reporting**

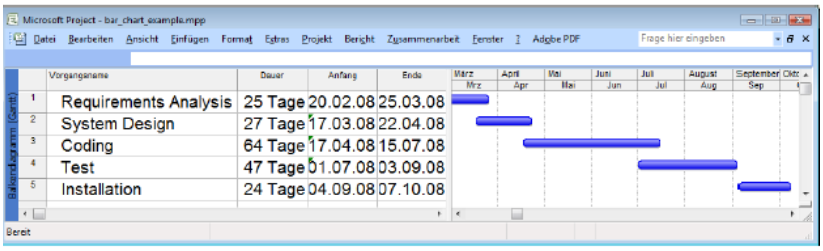
  - Bar chart
    - shows activities, start date and end date (or duration)
    - shows schedule against calendar time
    - Purpose: **reporting** and **progress tracking**



  - Activity network diagram
    - same as bar chart
    - shows dependencies between work packages and workflow
    - used to calculate critical paths
    - Purpose: **planning**

- Milestone chart
  Simple communication means between **customer** and **developer organization**
  就是按顺序，重要 activities 的结项时间（单个时间点，没有长度），单条排下来
- Bar chart
  Communication means between customer and project management
  - Is a transformation of a milestone chart with added durations
  - Work packages are grouped
    - phases, sub-projects
  - Dependencies must be known, but are not visualized explicitly
  - Temporal overlapping of work packages can be recognized immediately

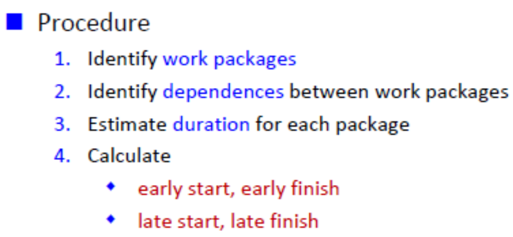
  就是 milestone 的扩展，加上 duration，而且知道各种 dependency，但是并没展示出来



Microsoft Project - bar_chart_example.mpp
Datei Bearbeiten Ansicht Einfügen Format Extras Projekt Bericht Zusammenarbeit Fenster ? Adobe PDF
Frage hier eingeben
Balkendiagramm (Gantt)
März Mrz
April Apr
Mai Mai
Juni Jun
Juli Jul
August Aug
September Sep
Okto
Bereit

	Vorgangsname	Dauer	Anfang	Ende
1	Requirements Analysis	25 Tage	20.02.08	25.03.08
2	System Design	27 Tage	17.03.08	22.04.08
3	Coding	64 Tage	17.04.08	15.07.08
4	Test	47 Tage	01.07.08	03.09.08
5	Installation	24 Tage	04.09.08	07.10.08

- Activity network diagram
  Identification and illustration of critical paths and floats



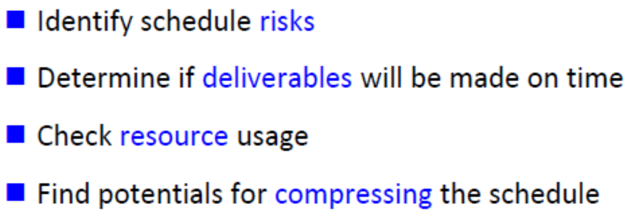
- Procedure
  1. Identify work packages
  2. Identify dependences between work packages
  3. Estimate duration for each package
  4. Calculate
     - early start, early finish
     - late start, late finish

No loops and no dangling arrows

四种 precedence relationships 都用在这里，计算 ES, EF, LS, LF

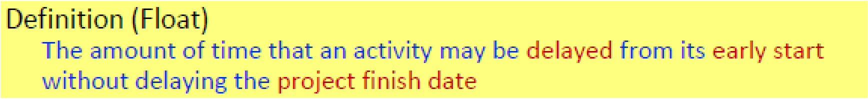
1) Finish-start dependency
2) Start-start dependency
3) Finish-finish dependency
4) Start-finish dependency

- ES, EF 用 forward calculation, LS, LF 用 backward calculation
- 对于一个 schedule，我们应该



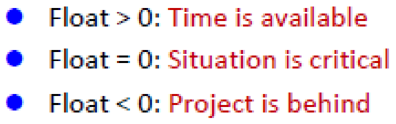
- Identify schedule risks
- Determine if deliverables will be made on time
- Check resource usage
- Find potentials for compressing the schedule

- Float



Definition (Float)
The amount of time that an activity may be delayed from its early start without delaying the project finish date

针对单个 activity 说的

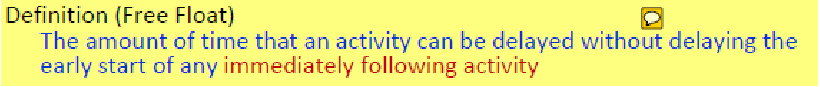


- Float > 0: Time is available
- Float = 0: Situation is critical
- Float < 0: Project is behind

- Critical work package:

Those float≤0

- Free float



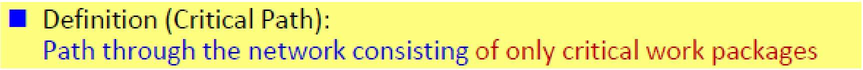
Definition (Free Float)
The amount of time that an activity can be delayed without delaying the early start of any immediately following activity

比 float 要求更苛刻

float 是不影响最后的结果，自己不在 critical path 上即可

free float 是完全不影响后面的 es，也就是每个后辈都有其他的瓶颈

- Critical path

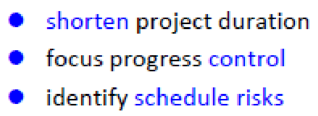


- Definition (Critical Path):
Path through the network consisting of only critical work packages

一条从头到尾的通路，且全是 critical work package， float 和为 0

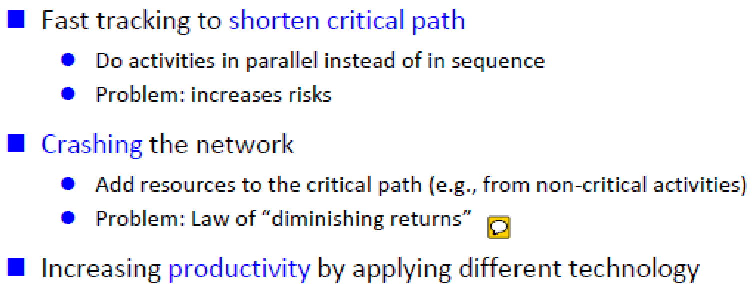
可以有多条 critical paths

Critical path is important for:



- shorten project duration
- focus progress control
- identify schedule risks

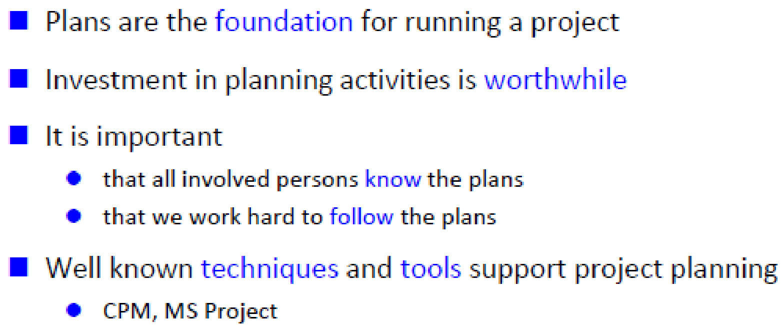
- Any critical work package 都在某一条 critical path 上面
- Schedule compression – methods



- Fast tracking to shorten critical path
  - Do activities in parallel instead of in sequence
  - Problem: increases risks
- Crashing the network
  - Add resources to the critical path (e.g., from non-critical activities)
  - Problem: Law of “diminishing returns”
- Increasing productivity by applying different technology

Diminishing returns: 收益递减，也就是回报不够投入的多

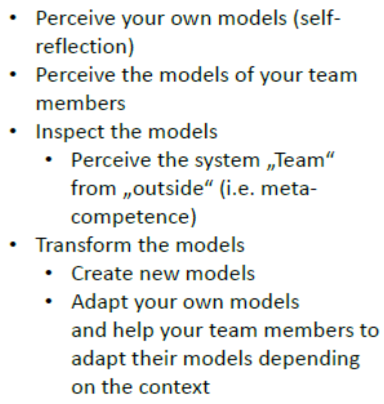
- Over time and weekends should not be considered during planning
- Summary:



- Plans are the foundation for running a project
- Investment in planning activities is worthwhile
- It is important
  - that all involved persons know the plans
  - that we work hard to follow the plans
- Well known techniques and tools support project planning
  - CPM, MS Project

11 Team leadership

- Basic assumptions on team leadership
  1) Social Factors determine 7/8 of the success factors
  2) Success criteria can be influenced only indirectly via the success factors.
  3) We can only perceive those (e.g. objects, humans) for which we have a model
- Key activities of leadership



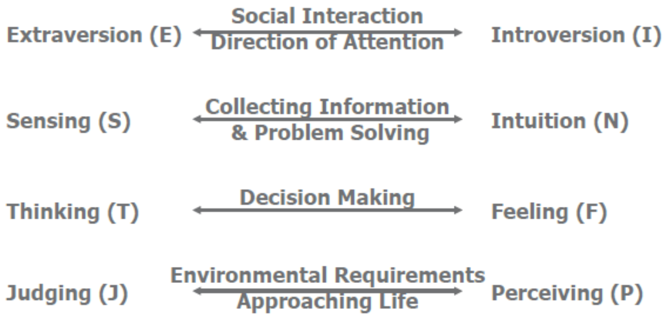
- Perceive your own models (self-reflection)
- Perceive the models of your team members
- Inspect the models
  - Perceive the system „Team“ from „outside“ (i.e. meta-competence)
- Transform the models
  - Create new models
  - Adapt your own models and help your team members to adapt their models depending on the context

认清自己；认清队友；查看模型；改变模型

- Personality model:

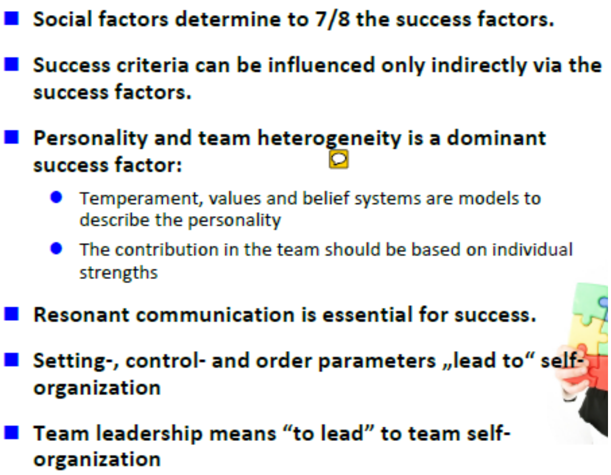
1) Temperament
2) Belief system
3) values

- Temperament: MBTI dimensions


Extraversion (E) ← Social Interaction / Direction of Attention → Introversion (I)
Sensing (S) ← Collecting Information & Problem Solving → Intuition (N)
Thinking (T) ← Decision Making → Feeling (F)
Judging (J) ← Environmental Requirements / Approaching Life → Perceiving (P)

MBTI typology 类型学: e.g. ENTP

- Basic assumptions (belief systems) are mental models which generalize made experiences to give orientation in the search of sense.
  e.g. Innovation results from customer wishes.
  感觉就是一条条信念，想法
- Values (moral concepts): are generalized, emotional guidelines in the search of sense an are derived by generalization from concrete experiences. Values are norms for social action and the basis for the cohesion and further development of a society or organization.
  e.g. love, friendship, duty...
  感觉就是我们认知的价值
- Self-organization
  Tuckman phases:
  1) Forming
  2) Storming
  3) Norming
  4) Performing
  5) Transforming/adjourning
- Summary:

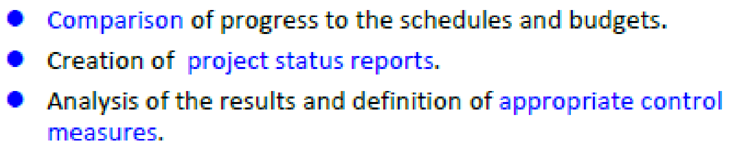


- **Social factors determine to 7/8 the success factors.**
- **Success criteria can be influenced only indirectly via the success factors.**
- **Personality and team heterogeneity is a dominant success factor:**
  - Temperament, values and belief systems are models to describe the personality
  - The contribution in the team should be based on individual strengths
- **Resonant communication is essential for success.**
- **Setting-, control- and order parameters „lead to“ self-organization**
- **Team leadership means “to lead” to team self-organization**

Team leaderships means to tune the parameters.
Resonant: 共鸣的，heterogeneity: 异质的

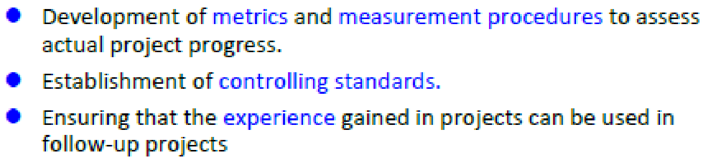
12 Monitoring and controlling

- Primary goals:



- Comparison of progress to the schedules and budgets.
- Creation of project status reports.
- Analysis of the results and definition of appropriate control measures.

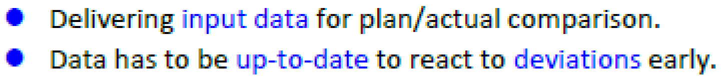
Secondary goals:



- Development of metrics and measurement procedures to assess actual project progress.
- Establishment of controlling standards.
- Ensuring that the experience gained in projects can be used in follow-up projects

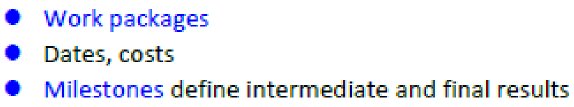
其实 primary 就是针对当前项目来说，要得出比较结果，然后找到 control measure
Secondary 就是对多个项目来说，总结 measure 的度量和过程，创建控制标准，积累经验

- Planning→
realization & control measures→
measure progress→
plan/actual comparison→
analysis→
corrective control measures 循环到 2
- Measure progress (上面的第三条)
Goal:

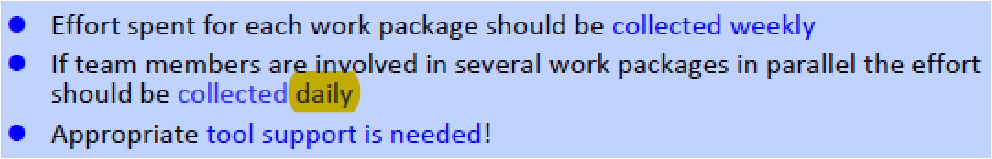


- Delivering input data for plan/actual comparison.
- Data has to be up-to-date to react to deviations early.

Unit of control:



- Work packages
- Dates, costs
- Milestones define intermediate and final results



- Effort spent for each work package should be collected weekly
- If team members are involved in several work packages in parallel the effort should be collected daily
- Appropriate tool support is needed!

上面的意思就是 measure 要用好的工具，非常勤快的 monitor

- Degree of completion – 3 methods
- Naive approach (我花费的 effort 和原计划的 effort 的比值):

$$DoC_{ES} = \frac{actualEffort}{plannedEffort}$$

不过这个方法是建立在错误的假设上:

- The planned effort is correct.
- If we have spent the effort we are done!

- Estimate to complete (ETC) : effort still needed to complete the work package
  - The value of the work that is already done is assessed
  - If the forecasts are totally wrong this approach has no use, too!

  本质就是估计剩下的工作量，和上面用老 plan 不同，这是新估计，但如果估计根本不准，新估计也没用。根据 forecast，我们需要 re-planning，而老的 plannedEffort 并不改变

  actualEffort = forcasetFinalEffort - estimateToComplete

$$DoC_{ETC} = \frac{actualEffort}{forcastFinalEffort}$$

- Earned value (有时不知道自己 actualEffort 或没法估计最后总 effort，用这个)

$$EarnedValue = plannedEffort - estimateToComplete$$

$$DoC_{EV} = \frac{earnedValue}{plannedEffort}$$

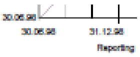
  Value 可以为负，也就是 ETC 比 plannedEffort 大

- EAC estimate at completion, total cost at end

  ETC estimate to completion, cost remain
- Degree of completion of work package

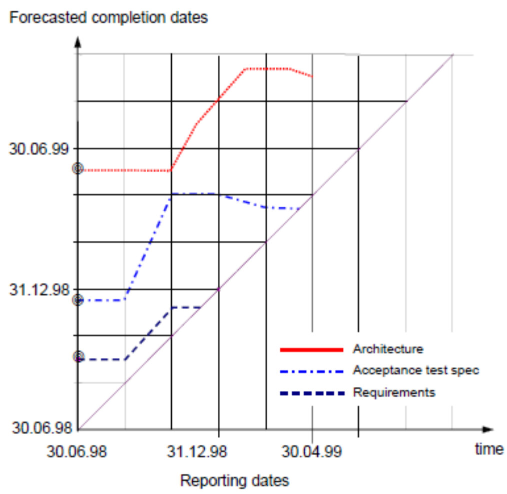
$$DoC = \frac{completedWP}{totalWP}$$

- Milestone trend analysis



  - vertical:
    - forecasted completion dates of all milestones
  - horizontal:
    - reporting dates (months, weeks)
  - Each milestone has a special symbol
  - At each reporting date the forecasted completion dates are added

  就是在每个 reporting date 更新 milestone 的预期日子，一个 milestone 一根线，线撞 45 度线时就是完成了



Horizontal: 保持不变

Ascending: 上扬，也就是进度不如预期

Descending: 将会早完成

- Earned value analysis (EVA)之前用 effort，这里用 value 和 cost
  1) Scheduling efficiency: actual/planned duration comparison
  2) Cost efficiency: actual/planned cost comparison

下面介绍 3 线

- PV (planned value), i.e. Budgeted Cost of Work Schedule (BCWS)就是计划的 budget，也就是 baseline，一条增长的线段

  Budget at completion (BAC) = PV(at the end)

- AC (actual cost)

  Cost incurred, i.e. Actual Cost of Work Performed (ACWP)

  Contains both **direct** and **indirect** cost

- EV (earned value)

  Planned cost for work completed

$$EV = DegreeOfCompletion * BAC$$

- Earned value analysis – method

  Express effort, cost and time as monetary value

  Compare the 3

- Cost variance: $CV = EV - AC$

  Schedule variance: $SV = EV - PV$

  Cost performance index (CPI): $CPI = \frac{EV}{AC}$

  Schedule performance index (SPI): $SPI = \frac{EV}{PV}$

  CPI and SPI both 1.0 is the best(optimal) case

  But there can be 1.23, 1.23, which is better than optimal

- 使用 CPI 和 SPI 分别可以重新估计 schedule 和 cost

  Estimate at complete (cost)

$$EAC = \frac{BAC}{CPI}$$

Forecasted duration

$$ForcastedDuration = \frac{plannedDuration}{SPI}$$

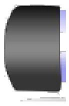
- To complete performance index (TCPI)

$$TCPI = \frac{BAC - EV}{BAC - AC}$$

就是要想用剩下的钱完成剩下的任务，需要达到多高的 efficiency，或者说 CPI

- Golden rules of earned value

  - Rule 1
    - Earned value should be verified by physically examining the work product associated with the work packages!
  - Rule 2
    - For unfinished work packages, earned value estimates are usually just a guess!



第二条用 3 种规则之一，50/50, 20/80, 0/100。一定要只用一种规则。

- Control measures:
  - corrective measures
    - goal: closing the gap between actual state and plan
  - plan modifications
    - goal: adapting the plan to the actual state of the project

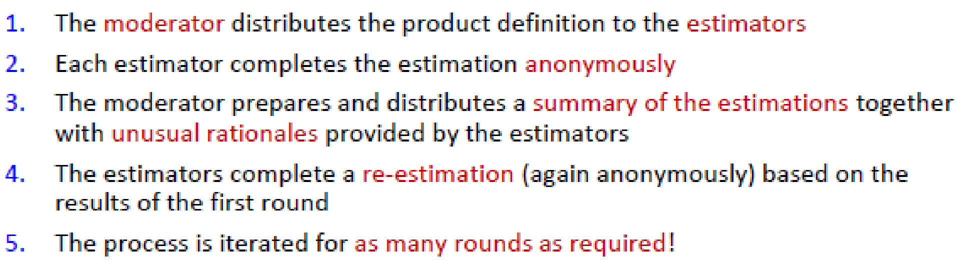
Select measure 时我们要 take into account the effects of the deviations to project goals

- The customer has be to informed.
- The customer has to agree with the control measures
  - reduction of functionality; reduction of quality
- We always have to check to what extent we can modify the projects goals (deliverables, costs, dates)

- Summary
  - Project monitoring and control is a continuous activity
  - Monitoring and control without plans is impossible
  - Plan/actual comparison is a prerequisite to define appropriate control measures
  - We have to collect the actual performance data
  - The units of control are work packages, costs, and the schedule
  - EVA offers a small set of proven monitoring metrics
  - Control measures have to be defined explicitly

13 Planning cost estimation

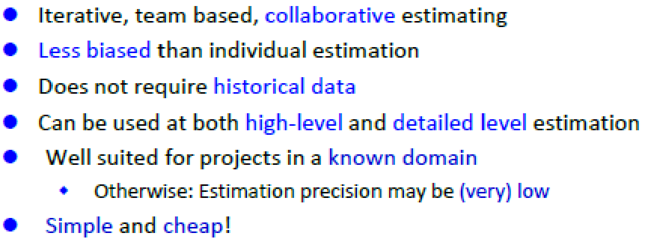
- Non-algorithmic methods
  1) Analogy costing method： 看类似的来
  2) Expert judgment： Delphi technique
  3) Bottom-up: estimate component 再加起来
- Delphi technique:



1. The moderator distributes the product definition to the estimators
2. Each estimator completes the estimation anonymously
3. The moderator prepares and distributes a summary of the estimations together with unusual rationales provided by the estimators
4. The estimators complete a re-estimation (again anonymously) based on the results of the first round
5. The process is iterated for as many rounds as required!

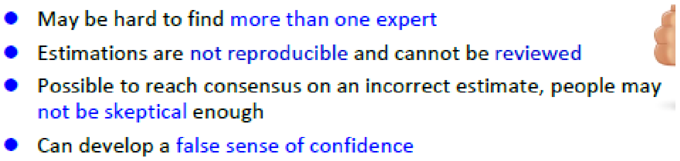
全程 anonymous，只有 moderator 做收发任务

Pros:



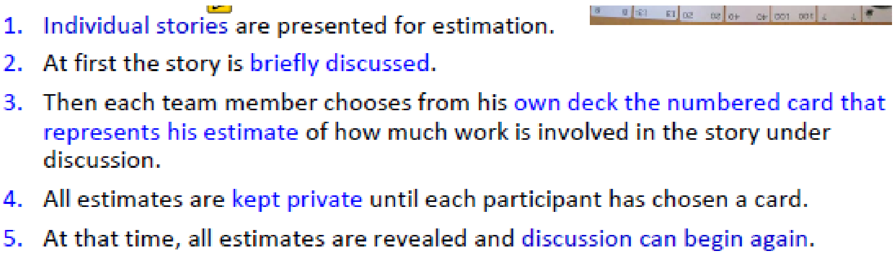
- Iterative, team based, collaborative estimating
- Less biased than individual estimation
- Does not require historical data
- Can be used at both high-level and detailed level estimation
- Well suited for projects in a known domain
  - Otherwise: Estimation precision may be (very) low
- Simple and cheap!

Cons:



- May be hard to find more than one expert
- Estimations are not reproducible and cannot be reviewed
- Possible to reach consensus on an incorrect estimate, people may not be skeptical enough
- Can develop a false sense of confidence

- Poker planning： agile variant of Delphi



1. Individual stories are presented for estimation.
2. At first the story is briefly discussed.
3. Then each team member chooses from his own deck the numbered card that represents his estimate of how much work is involved in the story under discussion.
4. All estimates are kept private until each participant has chosen a card.
5. At that time, all estimates are revealed and discussion can begin again.

Delphi, poker 都没说一定要 complete convergence

Story points may be map to person hours if you measure the team velocity for sprints, and they are constant

- Algorithmic estimation methods
  1) Function Point
  2) COCOMO
- Function Point:

  LOC is program size estimation, based on internal design, variant to programming languages

Function point estimation, based on external design, independent from languages

$$\text{Effort} = \frac{\text{Quantity} * \text{Quality}}{\text{Productivity}}$$

User Functions and groups of data 分五类

1) External input
2) External output
3) User inquiry
4) Internal data store
5) External data store

Counting rules:

1) Identify user functions
2) Classify to 1,2,3
3) Identify logical groups of data
4) Classify to 4,5
5) Assess

上面计算出来的是 unadjusted function points (uFP), i.e. quantity

Value adjustment factor (VAF) is quality

Based on 14 general system characteristics (GSC), each 0-5, base 0.65

$$VAF = 0{,}65 + 0{,}01 \cdot \sum GSC_i$$

$$aFP = VAF \cdot uFP$$

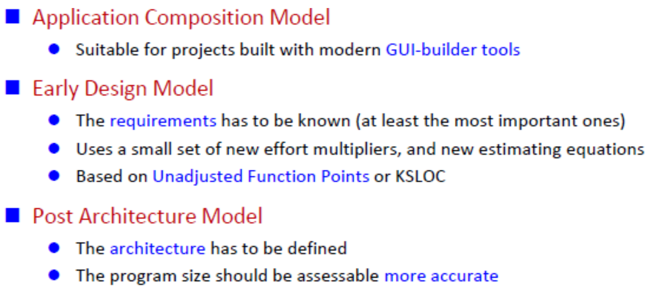
Convert FPs to effort:

1. 看经验的对应图
2. Convert by SLOC，就是看 1 个 function point 对应多少行代码的表，乘一下，然后用 cocomo 算 effort

- COCOMO (constructed cost model)

Input: program size LOC

适用的 model



■ Application Composition Model
- Suitable for projects built with modern GUI-builder tools

■ Early Design Model
- The requirements has to be known (at least the most important ones)
- Uses a small set of new effort multipliers, and new estimating equations
- Based on Unadjusted Function Points or KSLOC

■ Post Architecture Model
- The architecture has to be defined
- The program size should be assessable more accurate

$$Size = \left(1 + \frac{REVL}{100}\right) * (LOC_{new} + LOC_{equivalent})$$

REVL: requirements volatility 活泼性

$$Effort = A \cdot Size^{SE} \cdot \prod EM_i$$

SE (scaling exponent): 看五个的和*0.01+0.91(B)

1. Precedentedness (PREC)
2. Development flexibility (FLEX)
3. Risk resolution (RESL)
4. Team cohesion (TEAM)
5. Process maturity (PMAT)

EM (effort multiplier) 17 个，只用和自己有关的，连乘

Project duration,其中 Enom 是没有乘上 SCED 的 effort

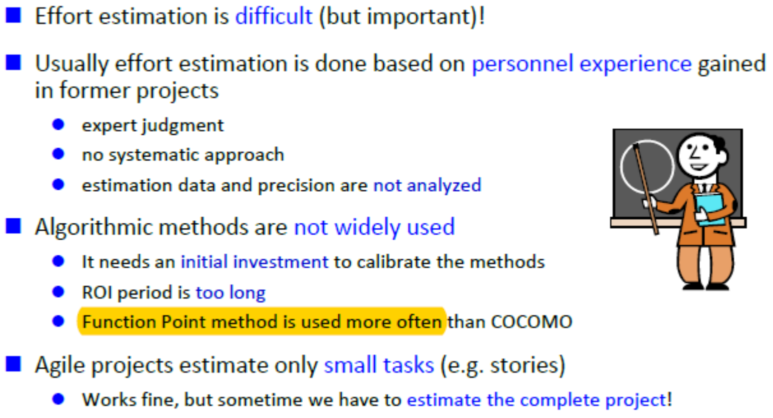
$$T_{nom} = C \cdot E_{nom}^{(D+0.2\cdot(SE-B))}$$

Staff:

$$AverageStaffing = \frac{Effort}{Duration}$$

如果新加 SCED,可以在原 duration 上面直接乘 0.75,0.85,1…

- Summary:



■ Effort estimation is difficult (but important)!

■ Usually effort estimation is done based on personnel experience gained in former projects

- expert judgment
- no systematic approach
- estimation data and precision are not analyzed

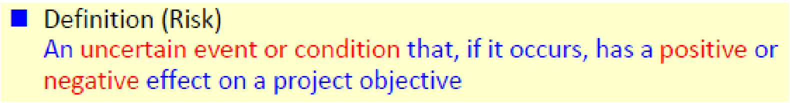
■ Algorithmic methods are not widely used

- It needs an initial investment to calibrate the methods
- ROI period is too long
- Function Point method is used more often than COCOMO

■ Agile projects estimate only small tasks (e.g. stories)

- Works fine, but sometime we have to estimate the complete project!

14 Risk management

- 

■ Definition (Risk)
An uncertain event or condition that, if it occurs, has a positive or negative effect on a project objective

可以是好的，也可以坏的影响

- Risk 3 components:

A possible event

Probability

Impact

- Risk classification by source (TPOLRE)

Technology

People

Organizational

Tools

Requirements

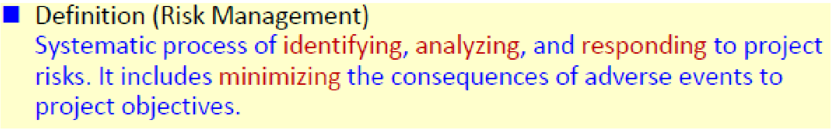
Estimation

- Risk classification by impact

Project

Product

Business

- 

■ Definition (Risk Management)

Systematic process of identifying, analyzing, and responding to project risks. It includes minimizing the consequences of adverse events to project objectives.

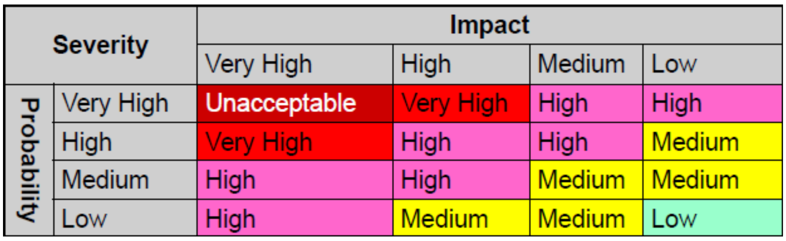
我们是 proactive management

Reactive management 就是 crisis management

- Risk management process
  1. Risk management planning
  2. Identification
  3. Analysis
  4. Response planning (mitigation)
  5. Monitoring and control, go to 2
- Identification

Checklist or SEI questionnaire (class, element, attribute 组成的 hierarchical)

- Analysis
  1. Probability
  2. Impact
  3. Severity



Severity		Impact			
		Very High	High	Medium	Low
Probability	Very High	Unacceptable	Very High	High	High
	High	Very High	High	High	Medium
	Medium	High	High	Medium	Medium
	Low	High	Medium	Medium	Low

- Response planning

Mitigation strategies

Avoid

Ignore/accept

Transfer

Contain/control

Establish contingency (set funds)

- Monitoring and control

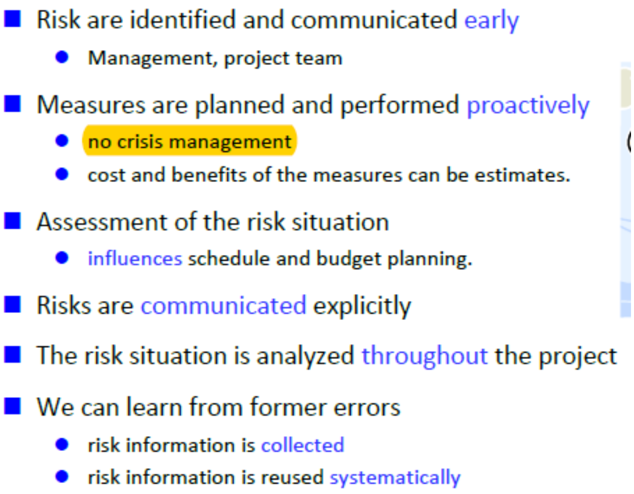
Re-assess risk situation

Implement, track, re-assess mitigation strategies, check if measures performed as planned

Communicate risk plan to stakeholders

Update documents

- Pros：



- Risk are identified and communicated early
  - Management, project team
- Measures are planned and performed proactively
  - no crisis management
  - cost and benefits of the measures can be estimates.
- Assessment of the risk situation
  - influences schedule and budget planning.
- Risks are communicated explicitly
- The risk situation is analyzed throughout the project
- We can learn from former errors
  - risk information is collected
  - risk information is reused systematically

Note

The book of maisic has solutions for some exercises after 523