The Software Lifecycle

Ant Kutschera
June 2011
HEIG VD

maxant

www.maxant.ch

Me

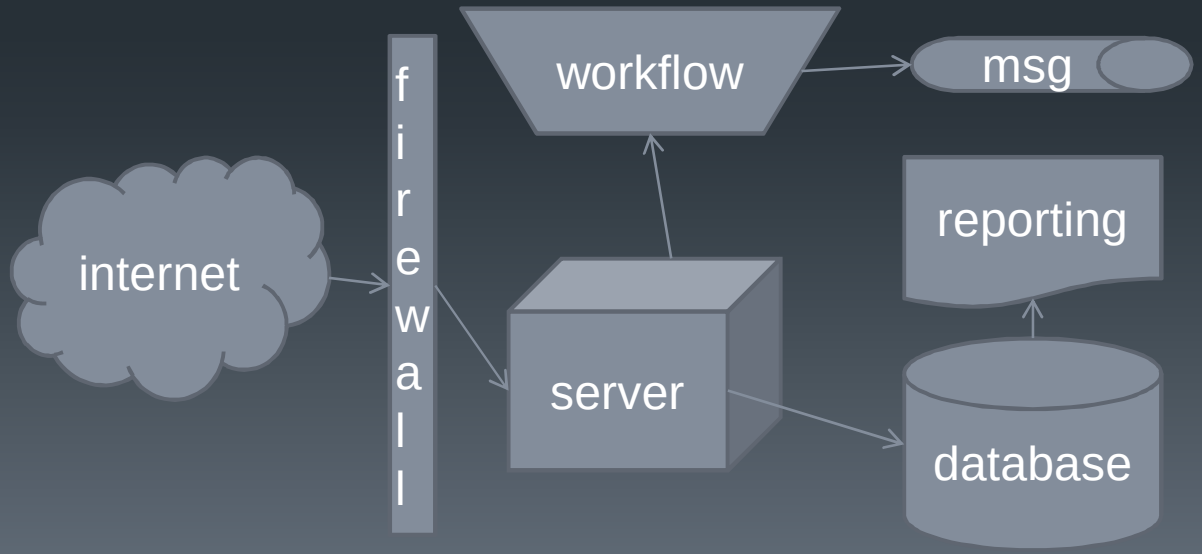
- Ant Kutschera
- Studied Aeronautical Engineering in the 90s
- Graduated with a PhD in Flight Simulation & Performance in 2000 from Loughborough University, Central England
- Joined a Software company in 2000
- Came to Switzerland in 2004
- Became Independent in 2006
- 11 years experience in Project Environments

- I am a Software Architect & Consultant with experience of
  - small projects – 1-3 developers, no project manager (PM)
  - medium projects – 3-5 developers, 1xPM
  - large projects – 5-15 developers, many PMs
  - programmes – 50 developers, 50 analysts, 20 managers

You

Architecture

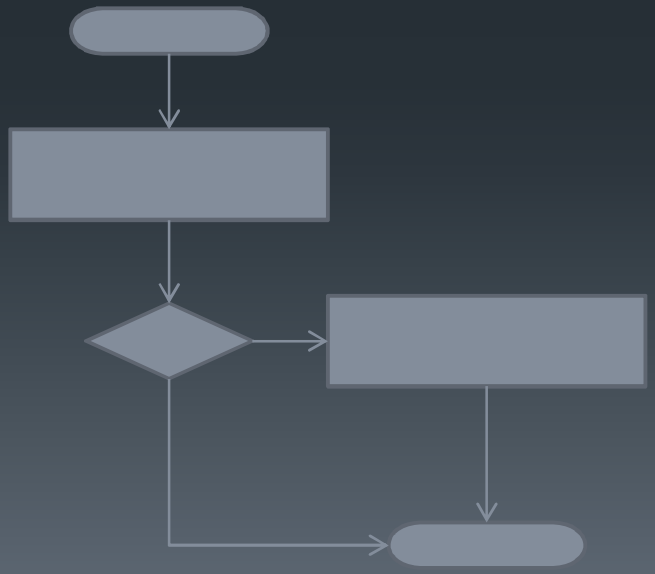
- Architecture is
  - high level design
  - trade offs between technology, resources, budget
  - strategic
    - how does software fit with the rest of the company?
    - technology choices
  - what components are there?
  - Paradigms: SOA? 3 Tier? Sequential/Parallel?

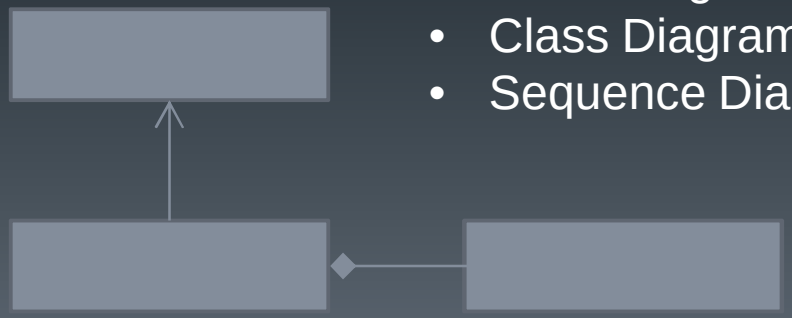


UML
- Package Diagrams
- Deployment Diagrams
- Component Diagrams

Design

- Interfaces between components
- What each component does
- Flow charts
- Database design
- Design patterns – model-view-controller, singleton, listeners, etc.





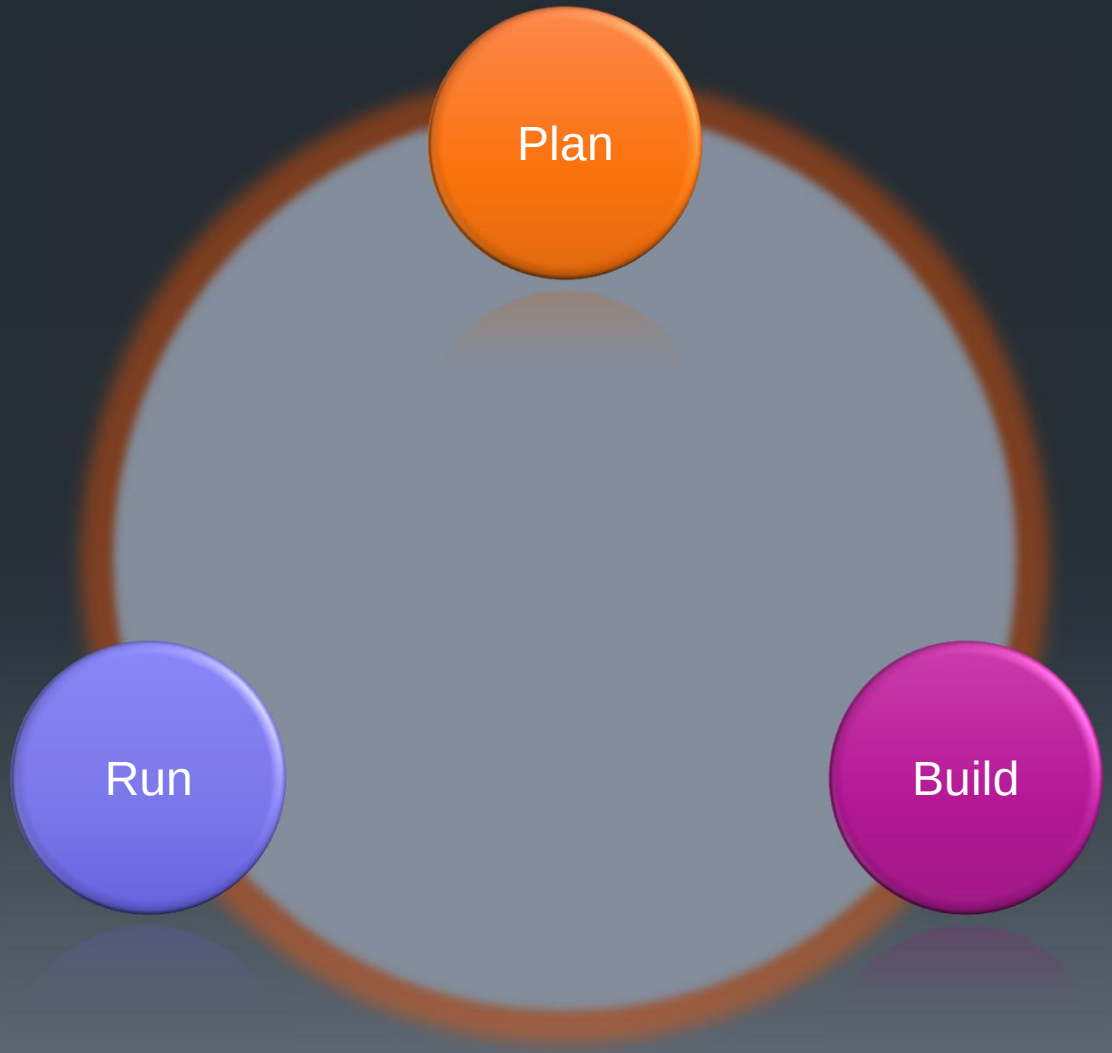
UML

- State Diagrams
- Class Diagrams
- Sequence Diagrams

Programming

- Individually
  - Think of your experiences of writing code for Mr. Chabloz
- As a team
  - Skills
  - Interfaces
  - “Getting run over by a bus”
  - Documentation
  - Communication
  - Collaborate
  - Learn to think differently

Software Lifecycle



Who is involved in Software Projects

- Stake Holders are the people:
  - with the vision (idea)
  - with the money
  - who see the business need for software
  - who will use the software
  - who will develop the software
    - Software Project Manager
    - Architects
    - Designers
    - Programmers
    - Testers
    - Documenters
    - Deployment
  - who test the software
  - who accept the software
  - who sell the software
  - who will maintain the software
- Anyone affected by the software
  - You and me, if the software is in a nuclear power plant and it fails!

Which Stakeholder?

- coordinates people
- facilitates meetings
- counts money
- handles politics
- organises things
- plans
- tracks progress
- motivates others

- This person is, or these people are:

Which Stakeholder?

- coordinates people
- facilitates meetings
- counts money
- handles politics
- organises things
- plans
- tracks progress
- motivates others

- This person is, or these people are:

Project Managers

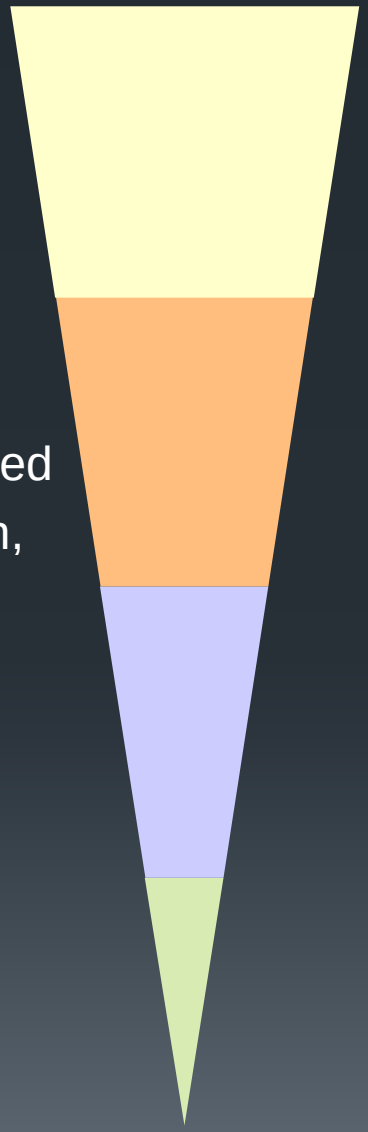
The role of a Project Manager

- To keep projects under control

- Planning
- Budgeting & Costs
- Hiring
- Making the team work well
- Tracking Progress
- Replanning & Fixing
- Re-budgeting
- Communicating
- Organising
- Firing
- Being a secretary
- Buying beer & donuts

Types of Project Manager

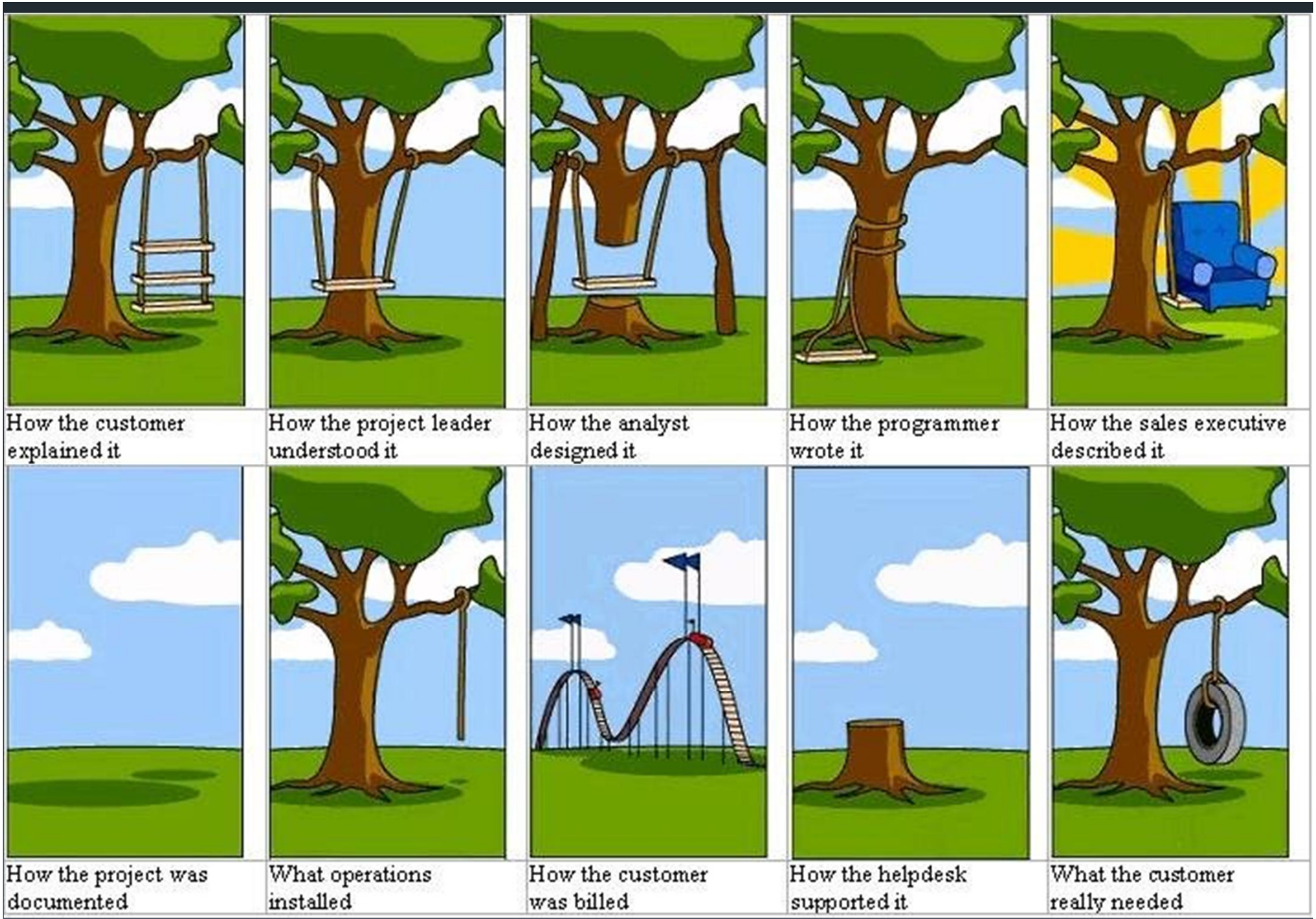
- Software Project Manager
  - Works for IT department, or IT consultancy
  - Must deliver project on time, on budget with the required functionality and quality
- Project Manager
  - Works for Customer, to ensure the *entire* project is delivered
  - Software, hardware, documentation, training, sales launch, launch party, overall budget, delivery date, etc.
- Programme Manager
  - Takes care of many projects for their company
- Programme Director
  - The Boss



How do you create software?

- It starts with a Vision
- Then someone with some money finds the vision useful
- They tell someone about the vision
- That person or people try to write it down in a Requirements Specification document
- Technical people analyse the requirements and decide how to build it in software. They write “Use Cases”
- Architects and Designers write Software Design documents
- Programmers write the software
- Testers test the software
- Someone decides that the software is OK and ready to ship
- The sales people might sell the software
  - e.g. Microsoft
- The customer uses the software

- *Where could it go wrong?*



How can a Project Manager avoid or fix these problems?

- Estimating
- Planning
- Building a good team
- Tracking & replanning
- Communicating and making sure people talk and work together
- Get more money & manage expectation
- Motivate the team
  - Beer and donuts!

Estimating

- Let’s try an exercise…
  - I want someone to build me some software which can:
    - Let the customer listen to MP3s
    - 1 million songs in the catalogue
    - 24/7 access
    - Customer register and login
    - Customer pays each month with credit card

Estimating

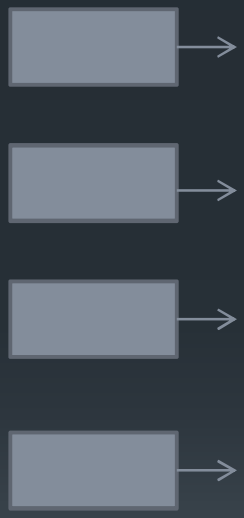
- Try to list all the things which need to be done
- For each thing, estimate the minimum and maximum number of days it will take one person to do it
- If a task is big, break it into small ones
  - A task should take 1 – 5 days only

Estimating

- Break it down into small tasks that you understand.
- Each task should take less than 5 days
- The people who will do the work should be involved in estimating the task
- Manage dependencies
  - + 30% ?
- Factor your estimate.
  - I use a factor of 200%
  - This is based on experience

Dependencies

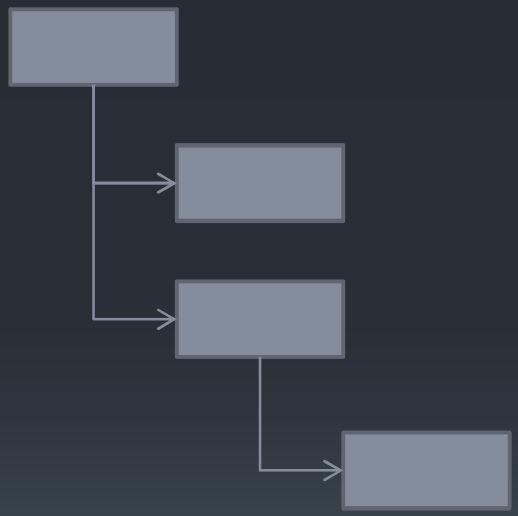
- Have you ever put up wallpaper?
  - + 20%, to match patterns?



4 man weeks in 1 week = 20k CHF

Dependencies

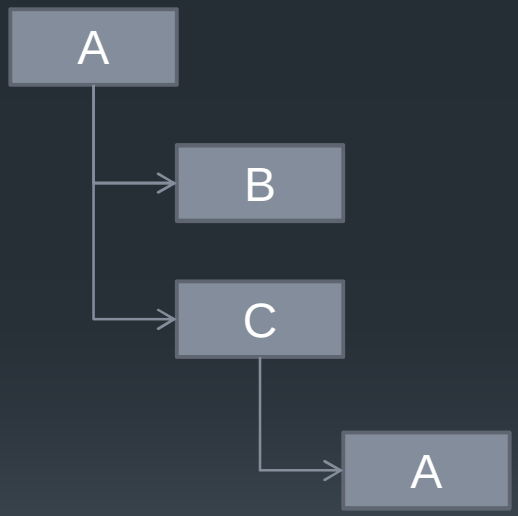
- Have you ever put up wallpaper?
  - + 20%, to match patterns



Dependencies: 4 man weeks in **3** weeks = 20k CHF, 200% delay

Dependencies

- Have you ever put up wallpaper?
  - + 20%, to match patterns



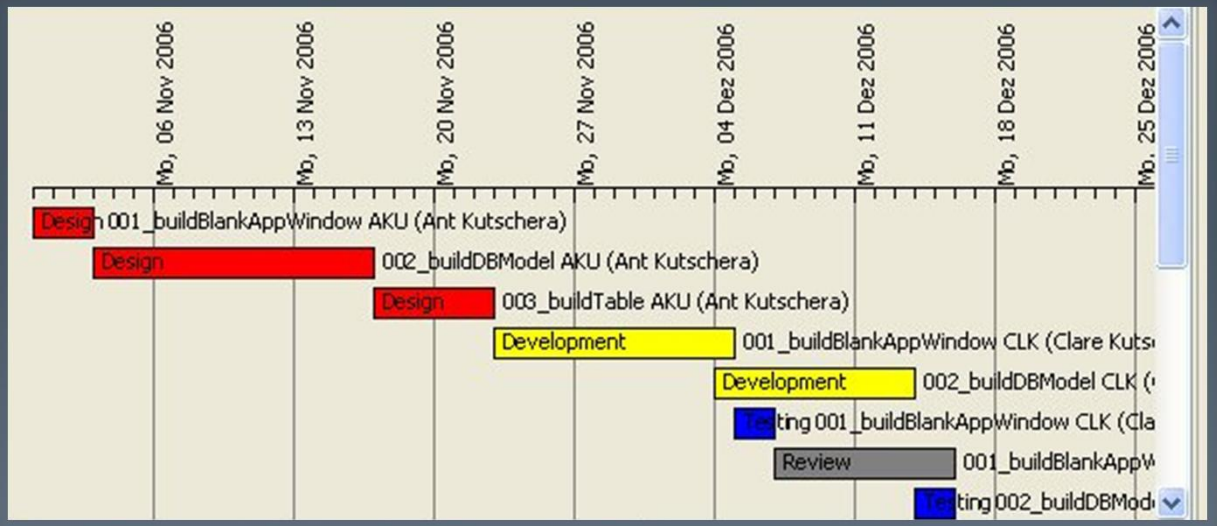
B idle	A idle
C idle	

Bench time: e.g. **7** man weeks in **3** weeks = **35k** CHF! 200% delay, 75% overspend

Planning

- Task dependencies
- “Resources” (people, machinery, etc.)
- Skills – who does which tasks?
- Milestones

- All are used to build a plan:



Build the Software

- We have the money
- We have a plan
- What now?

- Technical people need to:
  - Create the architecture
  - Design the software
  - Specify components
  - Create a test plan

- These things affect the plan
  - Planning is an iterative process

Build a team

- Where do the people come from?

- You might
  - get people from a specific department
  - hire temporary people
  - just be given people
  - use a consultancy
  - outsource
  - send it off shore

Tracking

- How do you measure success and failure?
- Up to 70% of software projects fail to some degree
  - http://www.it-cortex.com/Stat_Failure_Rate.htm
- Each task has:
  - Cost
  - Time
  - Functionality
  - Quality
- Measure these things and compare them to your plan

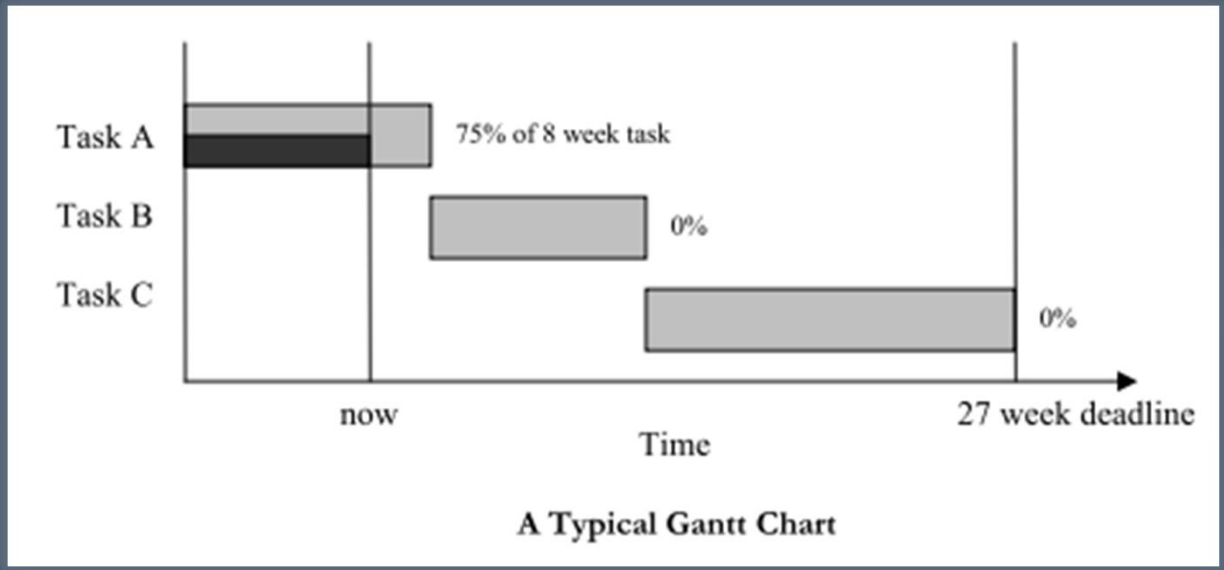
Tracking

- So how can you measure these things?

- Cost – timesheet each week, together with each person's hourly rate
- Time – Gantt Chart – are we on time? What is the critical path?
- Functionality – How many tasks are completed? How many still to go?
- Quality – How many bugs are open?

- "Velocity" is a term used in agile projects

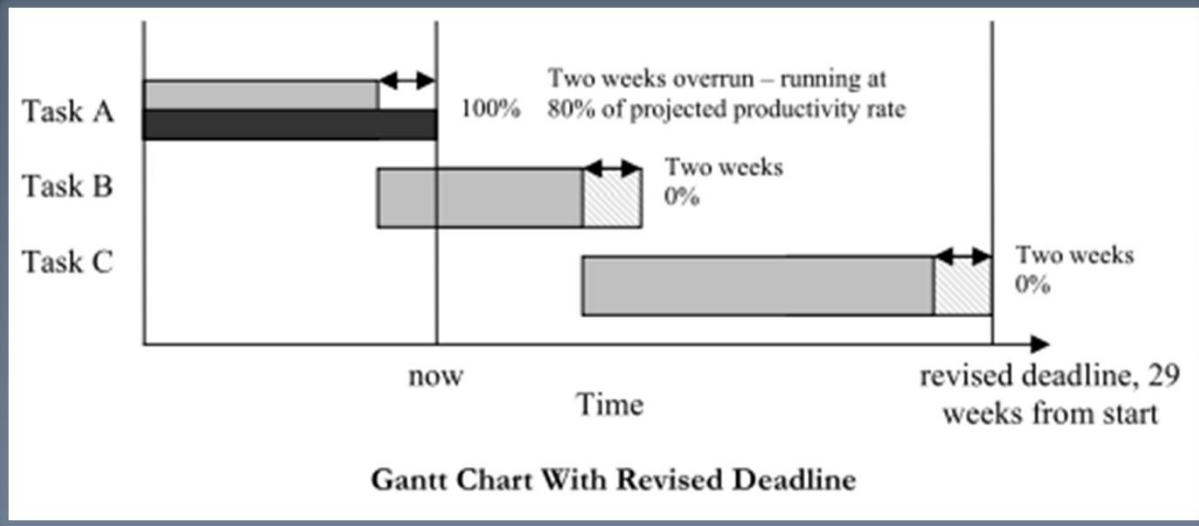
Velocity



A Typical Gantt Chart

Oh no!
We have a 2 week delay…

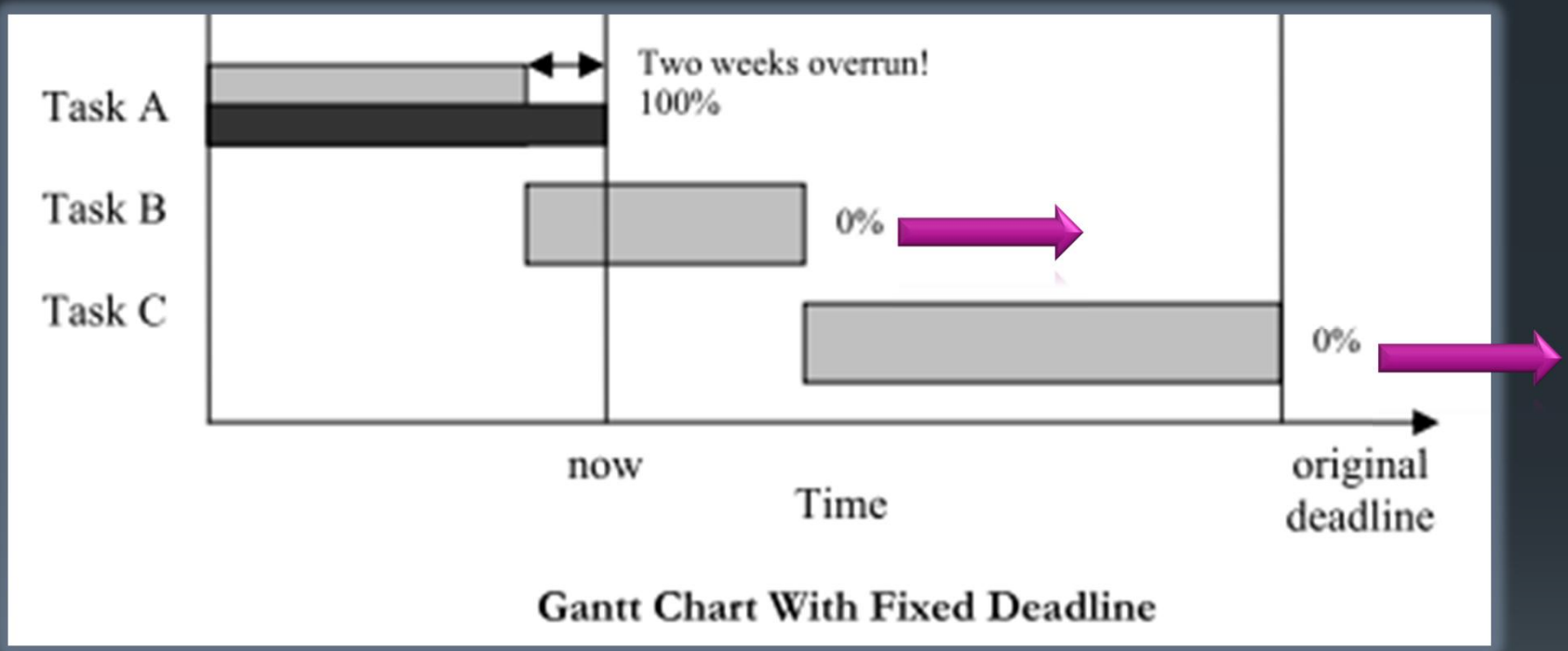
Will this work?



Gantt Chart With Revised Deadline

Velocity

Productivity must increase 40% above current rate in order to maintain deadline!

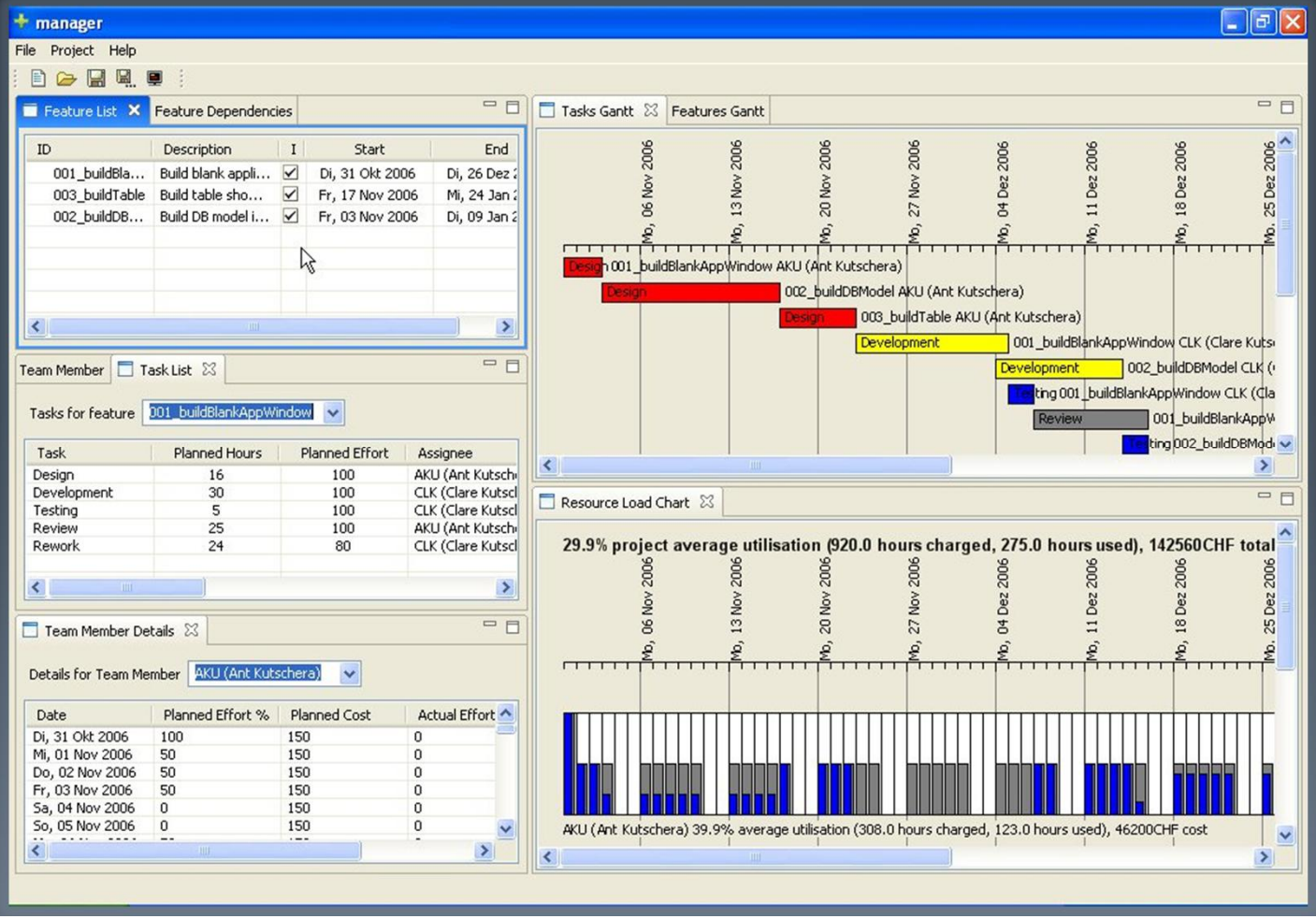


Gantt Chart With Fixed Deadline

It probably will NOT work! Velocity is the measure of how fast you are compared to the original plan

Replanning

- Measure progress. Add any new tasks. Replan.
- Do this every week.



Communication

- Weekly Reporting
- Show red / yellow / green
- Highlight risks
- Highlight successes
- Highlight current critical tasks

- All stake holders should know what is going on

Politics

- John is the project manager.
- Peter wanted to be the project manager, but didn't get promoted.
- Robert used to work with John but doesn't like him because he laughed at a joke about Robert once
- Peter wants to go out with Sarah
- Sarah is married
- John heard about a different project which is better for his career

- **Projects are sometimes like kindergarten!**

Iterative Build

- What is better?
  - Spend 6 months with 10 people building software and show the customer when it is ready
    - Sudden surprise?
  - Show the important people the software every week or two and keep checking it is what they need?
    - Scope creep?

Testing

- Which stakeholders will make sure the software does, what it is supposed to do?

- Programmers?
- Architects?
- Analysts?
- Project Manager?
- Testers?
  - IT department?
  - Customer?

- Testing ensures the required QUALITY is achieved

Testing

- Many levels
  - Programmer writes “unit tests”
    - White box tests; technical tests
  - Architect tests integration of components
    - White box tests; load testing / performance
  - (Business) Analyst tests
    - Black box, first round; does the software do what his analysis document says?
  - Software Tester
    - System testing; integration testing; black box; data tests; load testing / performance
  - Customer
    - Acceptance testing

Run

- After maybe 6 months, you have software that works!
- The customer has approved it
  - Acceptance Testing
- Is this the end?
- No!
- Software built today may run for 20 years
  - SBB/CFF online shop -> oldest code is 7 years already
  - SBB/CFF point of sales -> from around 1993
  - Microsoft Excel -> I bet there is code in there from one of the first versions - 20 years old? Excel is always backwards compatible
  - Linux -> some code over 15 years old?
  - Unix -> some code 30 or more years old!

Production & maintenance

- Maintenance budget can be more than it costs to build the software

- Call Centre
- Help Desk
- Support Line

- 1st, 2nd and 3rd level support
- Service Level Agreement – “SLA”

- On my project, every team member spends 20% of their time on maintenance – the project has been running for 4 years and we are still building new things
- On the “point of sales” software, they have 3 people working fulltime on 3rd level support

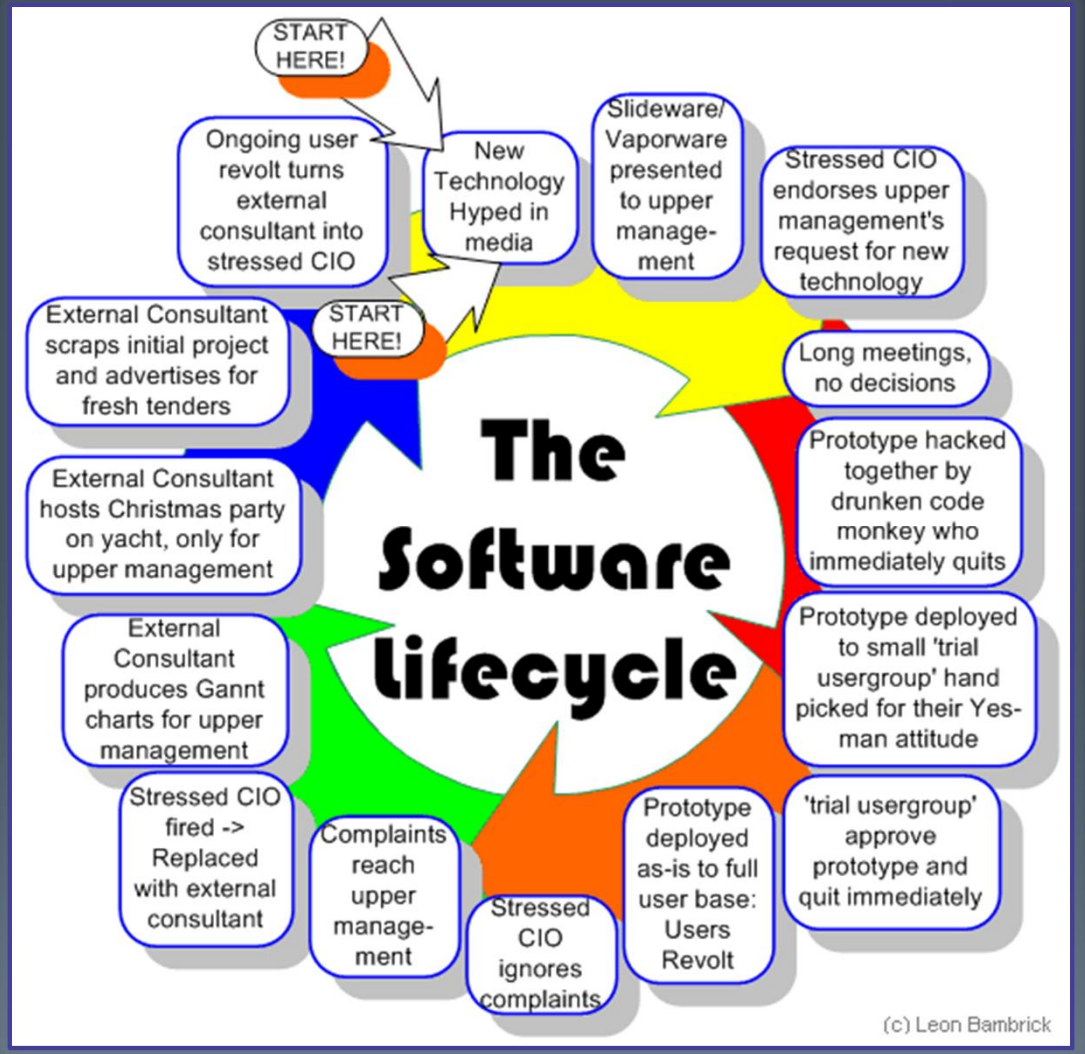
Failure

- What kinds of failure are there?
  - Project runs over budget
  - Project has poor quality
  - Project is late
  - Project does not do what is required
- Software Failure
  - Costs?
  - If amazon.com fails, how many dollars per second are lost?
  - If a nuclear power plant fails, how many lives are lost?
  - If an airplane crashes, how many people die?
  - If SBB/CFF cannot sell tickets, how many people have a bad day?
    - Cancellation task, cost a lot of money

Methodologies

- Waterfall / V-model
- Rational Unified Process
  - Keeps each team busy with next phase
- Agile
  - Feature Driven Development
  - Extreme Programming, XP
  - Scrum
- Others?
- Are they similar?

Questions



maxant

www.maxant.ch