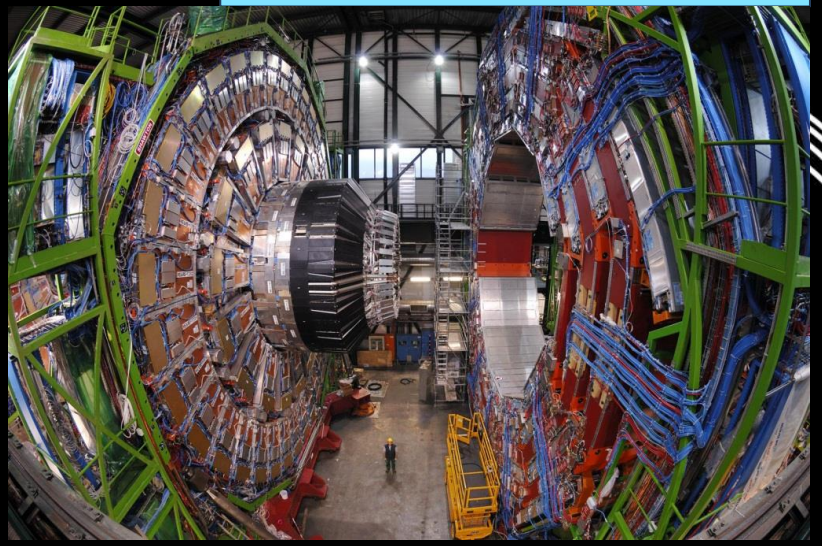
Let’s go engineering ….
Software Technology Master specialization

Software is eating the world

Marc Andreessen, Netscape founder

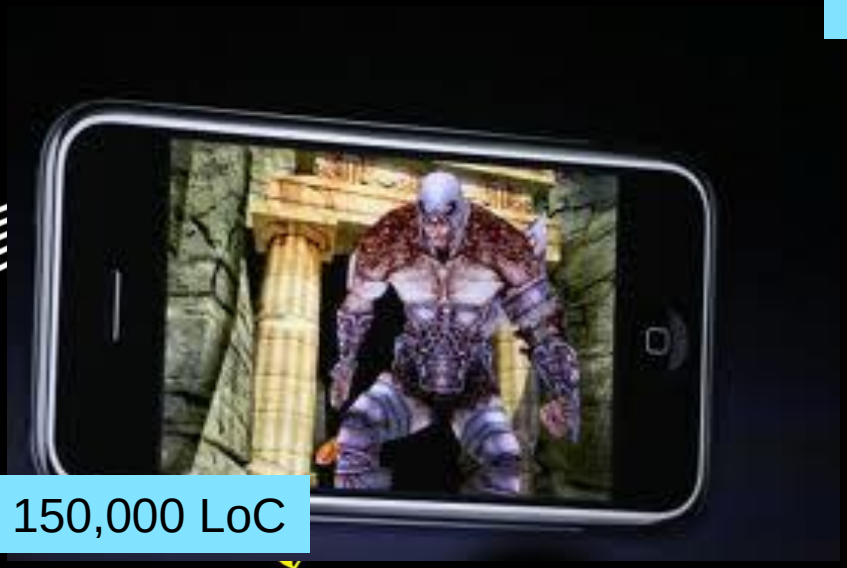
UNIVERSITY OF TWENTE.

CERN's Hadron Collidor
30,000 SW components



51,000,000 LoC

100,000,000 LoC

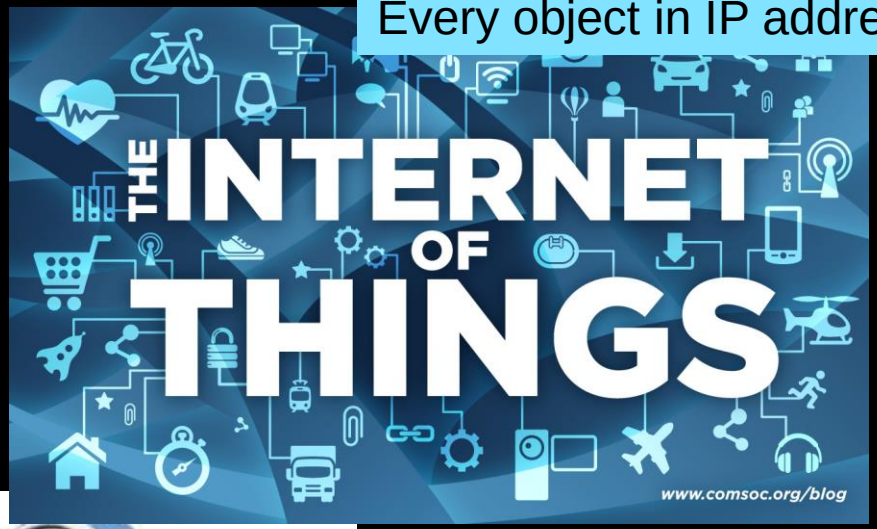


150,000 LoC

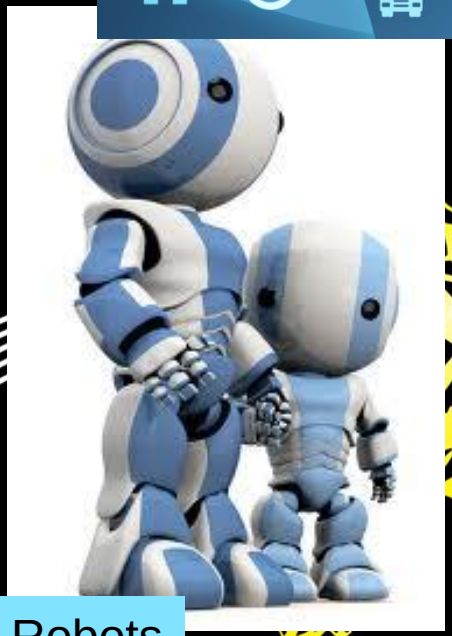
UNIVERSITY OF TWENTE.

This only gets worse ….

Every object in IP address



Self-driving cars



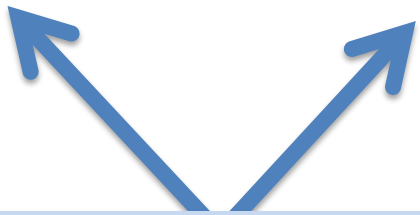
Robots

Wearable technology

Let's go engineering ….

more

- reliable
- maintainable
- secure
- scalable
- timely
- robust
- energy-efficient
- portable
- testable
- stable
- usable
- resillient



Write better software faster

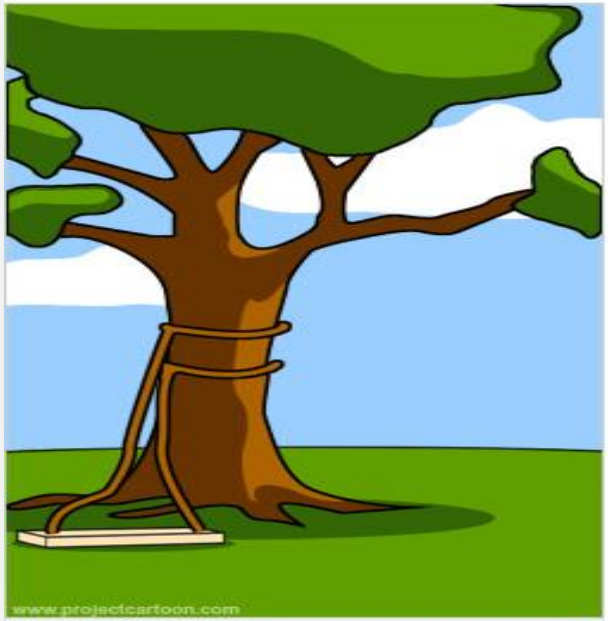
via

- model-driven engineering
- continuous integration / deployment
- novel programming languages / concepts
- code quality measurements
- domain-specific languages
- refactoring
- SOA, microarchitectures
- cloud computing
- multicore computing
- rigorous testing
- model checking
- …

How the customer explained it

How the project leader understood it

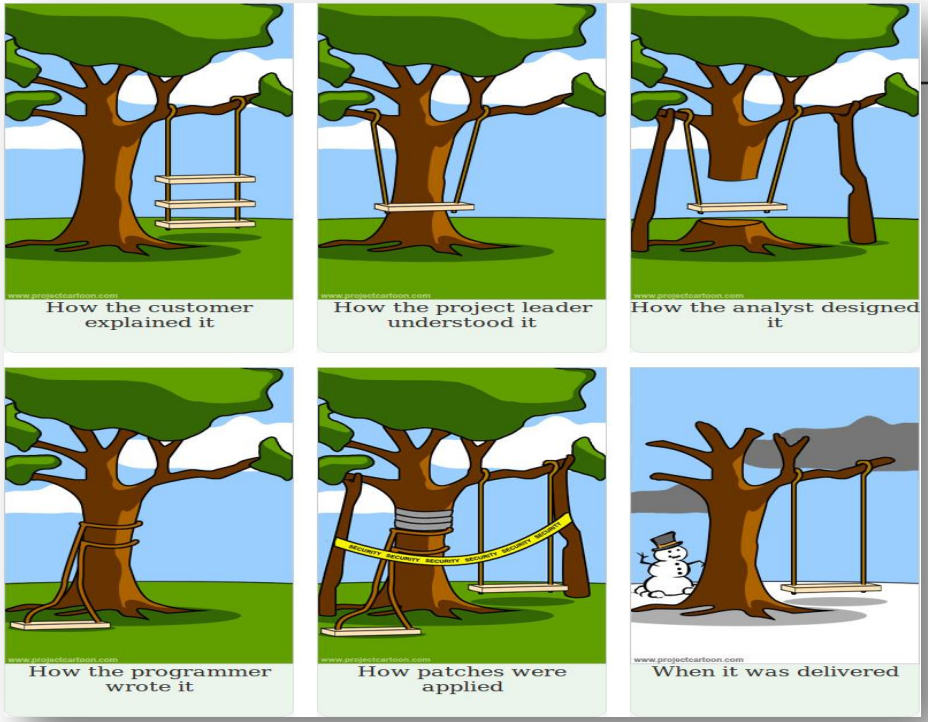
How the analyst designed it



How the programmer wrote it

How patches were applied

When it was delivered


How the customer explained it
How the project leader understood it
How the analyst designed it
How the programmer wrote it
How patches were applied
When it was delivered


www.projectcartoon.com
What the customer really needed

Who should take this Master programme?

- Are you interested in becoming a
  - **Software practitioner**
    … with theoretical knowledge and practical skills
  - **Researcher**
    … do theoretical and experimental research
  - **Tool builder**
    … proficient in state-of-the-art software engineering techniques

- During your Bachelor you have learned the **basic concepts and techniques** to work with Software Engineering

- Software Technology specialization educates you to **make a difference** in this area!

Highlights - 1

- **Broad coverage**
  - **Software Engineering phases**
    - requirements, architecture, design, …
  - **Technologies**
    - XML, parallel computing, programming languages, …
  - **Application areas**
    - security, cloud computing, real-time systems, …

- Includes courses from

TU Delft TU/e

Highlights 2

- Practical relevance
  - Industrial Software Engineering Project (**experience with real project**)
  - Industrial Advisory Board (**your future employer?**)
  - More practice, less theory

Kinds of final projects

- **Case Studies applying software technology**
  - On the Quality of Quality Models - J.H. Hegeman - @ Info Support BV
  - Evaluating the Behavior of Embedded Control Software – Christian Terwellen @ Océ
  - Towards Continuous Delivery in System Integration Projects - Sandra Drenthen @ Everett

- **Developing tools & Methods**
  - A Java Bridge for LTSmin - Ruben Oostinga
  - Trace-based debugging for Advanced-Dispatching Programming Languages – Marnix van ‘t Riet
  - Multi-Target User Interface design and generation using Model-Driven Engineering – Mark Oude Veldhuis @ Sigmax
  - Industrial Validation of Test Coverage Quality – Martijn Adolfsen

Curriculum Structure - Overview

- ***General***
  - Courses on fundamental SE concepts
  - 3 mandatory
- ***Phases***
  - Dedicated courses zooming in techniques from different SE phases
  - choose 3 from 6
- ***Technologies***
  - Courses applying different technologies
  - Choose 2 from 9
- ***Application Areas***
  - Software-related courses taken from different application areas
  - Choose 2 from 8
- **Electives**
  - up to 4 courses
- **Special courses**
  - upto 4 courses

Why should you take this Master programme?

- **Overlap with ISE specialization**
  - ISE: focus information and data technology
  - ST: broad education in software engineering
    room to specialize in technologies and application areas
    (other than IS)

- **Overlap with MTV specialization**
  - MTV: focus on quality assurance phase in SE process and on QA tools
  - ST: considering whole SE lifecycle

Curriculum Structure – General courses

3 mandatory

- **UT**: Software Management
- **UT**: ADSA – Model-Driven Engineering
- **UT**: Industrial Software Engineering project (*10 EC)*

Curriculum Structure - Phases

At least three of:

- **UT**: Specification of Information Systems (Requirements phase)
- **UT**: Testing Techniques (Quality Assurance phase)
- **UT**: Design of Software Architecture (Architecture phase)
- **UT**: Best Practices in Software Development (Detailed Design and Development phases)
- One of (Maintenance phase)**:**
  - **TU/e**: *2IS55* - Software Evolution
  - **TUD**: *IN4189* - Software Reengineering

Curriculum Structure - Technologies

At least two of:

- **UT**: System Validation
- **UT**: Modeling and Analysis of Concurrent Systems 1
- **UT**: ADSA – Product Line Engineering
- **UT**: Data Science
- **UT**: Concepts of Programming Languages
- **TU/e**: Advanced algorithms
- **TU/e**: Architecture of Distributed Systems
- **TUD**: Distributed Algorithms
- **TUD**: Parallel Algorithms and Parallel Computers

Curriculum Structure – Application Areas

At least two of:

- One of (application area “security”)
  - **UT**: Network Security
  - **UT**: Algebra & Security
- **UT**: Real-Time Software Development
- **UT**: Managing Big Data
- **UT**: Programming in Engineering
- **UT**: Wireless Sensor Networks
- **UT**: Cloud networking
- **TU/e**: Constraint programming

Curriculum Structure – Electives

up to four courses (to reach a total of 120 ECTS)

- **UT**: Advanced Logic
- **UT**: Capita Selecta Software Technology
- **UT**: Advanced Programming in Engineering
- **UT**: Advanced Requirements Engineering
- **UT**: Service-oriented Architecture Web Services
- **UT**: Graph Theory
- **UT**: Design Science Methodology
- **TU/e**: Algorithms for massive data
- **TU/e**: Geometric algorithms
- **TUD**: Embedded Real-Time systems
- And all courses from the Computer Science Master program at the University of Twente
- **A “Traineeship” cannot be chosen as part of the ST study package.**

Special courses

- ***Capita Selecta Software Technology*:**
- elective self-study course
- research on selected topics from Software Technology. Although not mandatory, we will recommend this course to all students with research-oriented interests.

- ***Industrial Software Engineering project*:**
- Project course, where teams develop a product. A company (eg from advisory board) acts as the client for this product. The team must follow a complete software engineering process.

- ***Best Practices in Software Development (Q 2B)***
  - Software patterns

- ***Concepts of Programming Languages (Q 1B)*:**
  - Progamming paradigma's

Local Embedding

- Relation to other specializations discussed before
- Most core courses from groups “Formal Methods and Tools” and “Services, Cyber security and Safety”
- Other groups contribute courses mainly in categories “technology” and “application areas”
- Students can do master project in any research group
  - Performing software engineering in the respective application domain
  - Develop or evaluate supporting tools

International Embedding - IEEE Software Engineering Body of Knowledge (SWEBOK)

Knowledge Area	Covered in Courses of SE Master Program
Software requirements	• Specification of Information Systems-
Software design	• Design of software architecture- • Advanced Design of Software Architectures – PLE- • Advanced Design of Software Architectures – MDE- • Service-Oriented Architecture with Web Services
Software construction	• Concepts of Programming Languages • Service-Oriented Architecture with Web Services • Best practices in software development-
Software testing	• Testing Techniques- • Best practices in software development-
Software maintenance	• Software Evolution- • Software Reengineering-
Software configuration management	-
Software engineering management	• Software Management-
Software engineering process	• Software Management-
Software engineering tools and methods	• Best practices in software development-
Software quality	• System Validation- • Modeling and Analysis of Concurrent Systems I-
Software engineering professional practice	• Industrial software engineering project
Software Engineering Economics	-
Computing Foundations	• Advanced algorithms • Parallel algorithms and parallel computers • Network security
Mathematical Foundations	• Algebra and security
Engineering Foundations	-

Details

- **Responsible chair**: Formal Methods and Tools
- **Homepage**: http://fmt.ewi.utwente.nl/education/st/
- **Program Mentor**: Prof. Arend Rensink (Zilverling 3090)
  a.rensink@utwente.nl