

Software Engineering

Reda Bendraou

reda.bendraou@lip6.fr

<http://pagesperso-systeme.lip6.fr/Reda.Bendraou/>

Software Engineering (SE)?

- “**Software engineering** is the application of a systematic, disciplined, quantifiable approach to the development and maintenance of software, and the study of these approaches; that is, the application of engineering to software”.
- “**Engineering** is the science, discipline, art and profession of acquiring and applying technical, scientific and mathematical knowledge to design and implement materials, structures, machines, devices, systems, and processes that safely realize a desired objective or inventions ”.

Source: Guide to the Software Engineering Body of Knowledge - 2004 Version. IEEE Computer Society. p. 1-1. ISBN 0-7695-2330-7. <http://www.swebok.org>.

Why is it so important?

Software is everywhere!

- Business processes (administrative etc.)
- Government
- Industry (factories, product lines)
- Transportations
- Defense, finance, healthcare...
- Medias...
- E-commerce...
- And much more!!

Economical and political issues/challenges

Why is it so important?

In case of bugs/ bad design, the consequences can be dramatic!

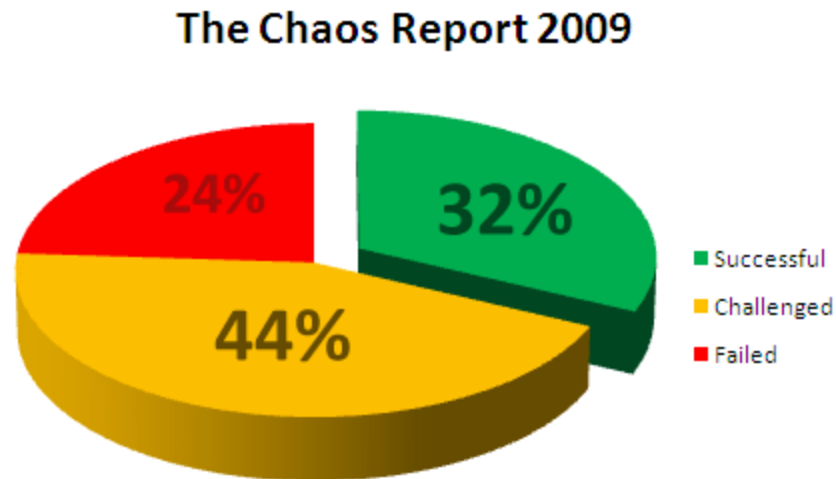
- Therac 25, '85' 87 : 6 patients were over exposed to x-rays, 2 patients died
- Denver Airport luggage distribution system, 2005: 16months, 560 M\$
 - http://calleam.com/WTPF/?page_id=2086
- Ariane 5 flight 88/501: 40s after takeoff , destruction., 850 M\$
- Mars Climate Orbiter & Mars Polar Lander : destruction.
- And much more....

SE's Achilles heel

- Cost
- Time
- Quality

Some Figures!

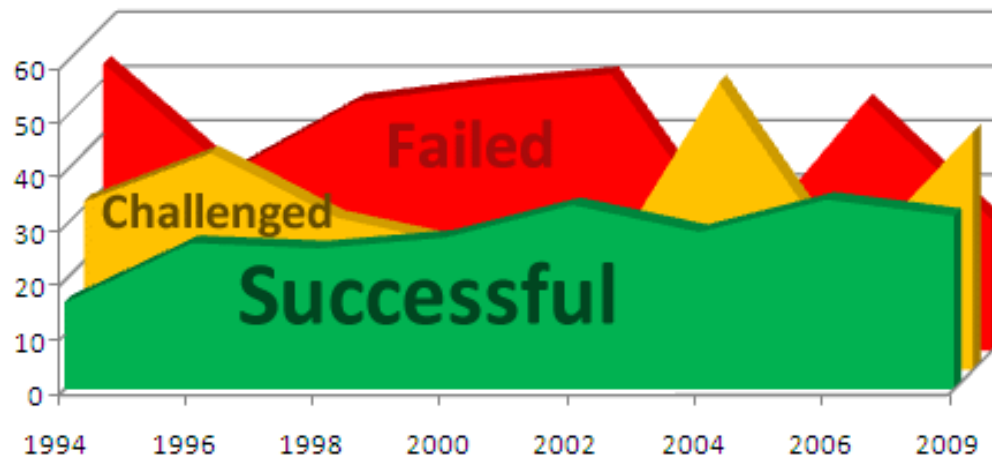
Source: The Standish Group



- Successful means on-time, on-budget, and with all features and functions as defined in the initial scope;
- challenged means late, over budget, and/or with less features and functions than defined in the initial scope;
- failed means cancelled prior to completion, or delivered but never used.

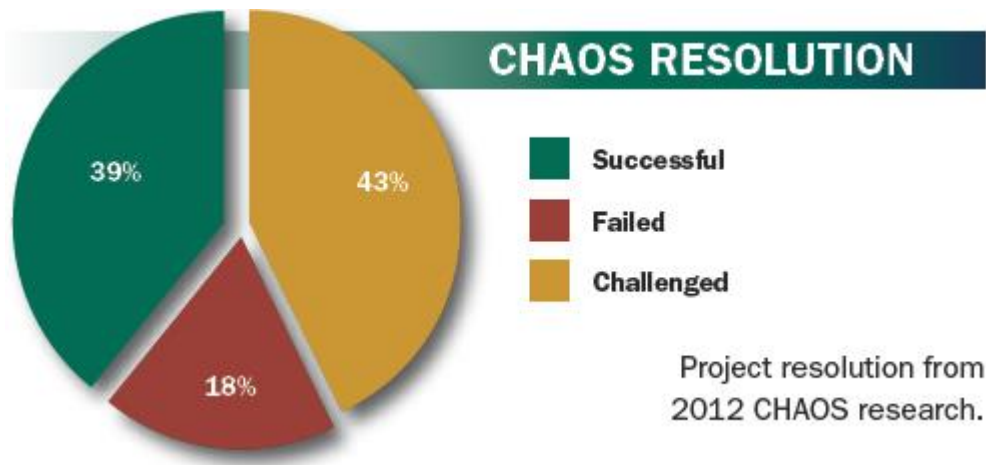
Some Figures!

- Success ratio constantly increasing, but failure ration remains unpredictable!



Other source, CHAOS manifesto'12

- <http://www.versionone.com/assets/img/files/CHAOSManifesto2013.pdf>



Issues related to SE

Issues related to the nature of software

- Have to constantly evolve, sometimes very often
- Requirements not always clearly expressed
- Sometimes, it's very hard to anticipate the impacts of a change/evolution
- The Copy/Paste problem !

Issues related to SE

Issues related to people

- Often, they don't know what they exactly need, or may express it in the wrong way
- Stakeholders may have difficulties to communicate with each others (from different domain, different vocabulary, etc.)
- Some software developers think that they know everything and better than the customer itself!

Issues related to SE

Technological issues

- Many languages and methods
- Constantly evolving, new versions, new tools, etc.
- Need to adapt in a short period of time

Why Software projects fail?

5 main reasons:

- Unrealistic commitments
- Bad project management
- Lack of means to control the project's progression (process)
- Use of inappropriate technology (methods, tools, languages)
 - Also called: the **Accidental Complexity**
- Not enough validation and verification

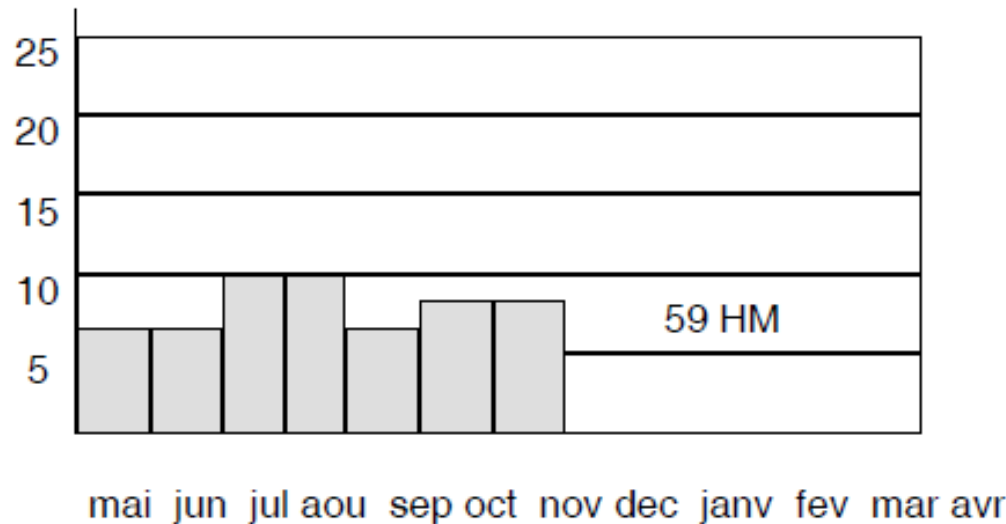
Example of a project (failure)

- Project in telecommunication(medium size/complexity)
- First experience of the team in this domain
- No means to evaluate the project's progression, no KPI (Key Performance Indicators) put in place
- A bad estimation of the project's complexity

Example of a project (failure)

Efforts needed (estimated at the beginning of the project): 59 M/M

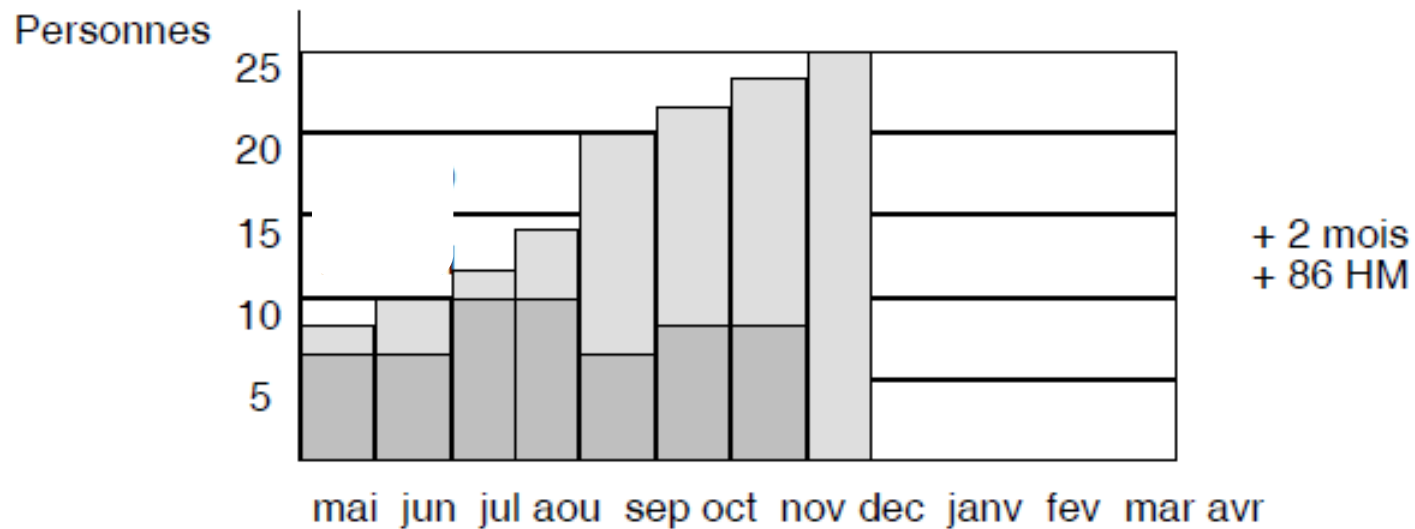
Personnes



Example of a project (failure)

5 months later

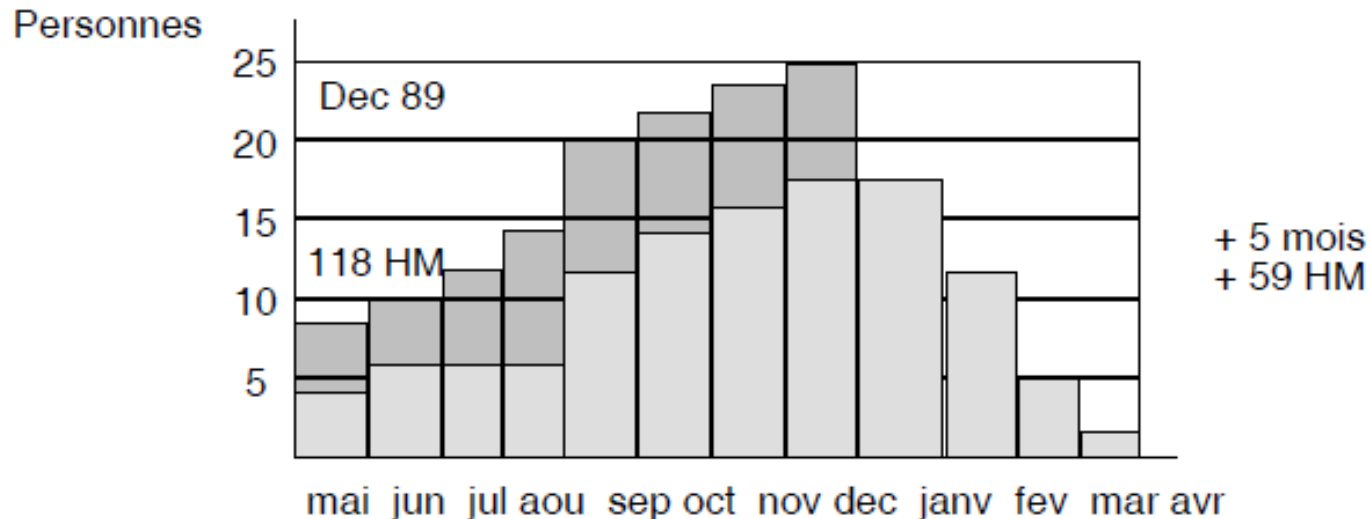
+ 2 months, +86 men/month



Example of a project (failure)

And 3 months later..

- + 5 months, +59 men/month
- No information about what has been implemented...
- The project is out of control
- The customer requires an Audit...



Example of a project (failure)

Analysis of the situation

- 23 sub-systems (modules) have been identified
- 67597 ± Lines of Code
- Situation after 8 months ...
 - 2 sub-systems Ok
 - 21 sub-system under design or coding

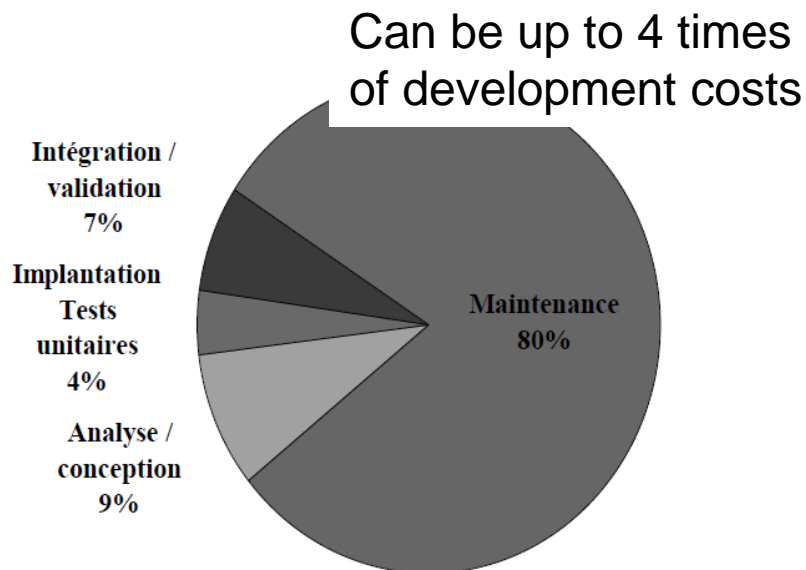
Example of a project (failure)

Results

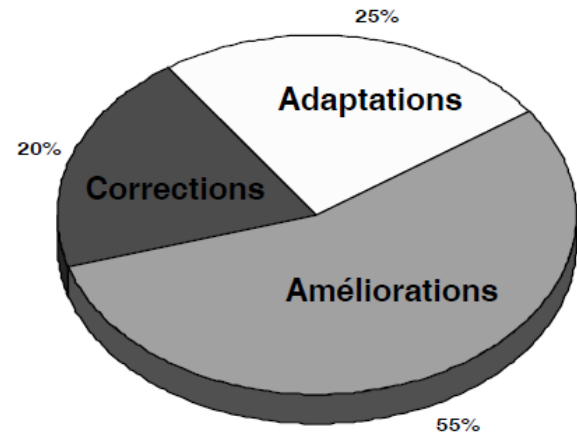
- Project duration reevaluated to 30 months (+18)
- Efforts reevaluated to 420 MM... (+300)!!!
- Budget increased : +4 500 000 \$

What about maintenance?

- **Project maintenance**
 - The software does not meet the customer's expectations!
 - Need for evolution, adaptation
- **An important part of the project's costs**



Types of Maintenance



So to sum up!

- **SE goals are:**
 - To improve quality
 - To optimize time
 - To reduce costs

- **3 main challenges SE has to face nowadays! => See next slides**

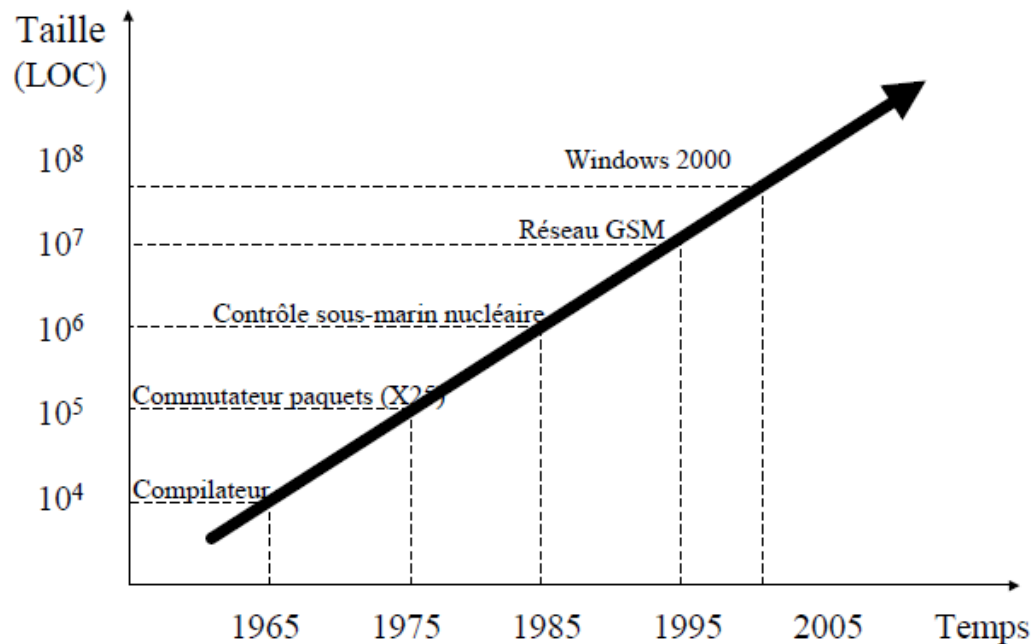
SE: challenges

- **Systems: bigger and bigger, distributed**
 - How to validate components (modules) developed separately
 - Inter-components validation /integration
 - Problems due to managing resources (human and means)
 - Hire the right skills, use the good tools, languages etc.
 - Project management
 - Use the right process, planning, offshore...



SE: challenges

- **Systems: bigger and bigger, distributed**
 - Linux Kernel 3.6 15M LOC
 - Mac Os, more than 86M LOC



SE: challenges

- **Components(modules) are more and more complex**
 - The problem of intra-component validation
 - It is still too much expensive to proof (verify) the code!
 - Only few people have expertise to do this
 - In practice
 - Dedicated to very critical components

SE: challenges

- **Requirements that constantly change/evolve**
 - Nokia reported that
 - 50% of requirements changed after the beginning of the project
 - 60% of these, changed at least 2 more times!
 - This is the usual situation, not an exceptional one
 - **When walking on water or developing software from a specification can be an easier task?**
 - if both are frozen (**Edward V. Berard**)

Any Solutions?

Some solutions addressed in the first part of this course

- **How to deal with complexity: (3 sessions)**
 - Modeling!
 - Model Vs. Code
 - UML: The Unified Modeling Language
- **For a better productivity: (1 session)**
 - Code generation & reverse engineering
 - Java as a coding language
 - The MDE (Model-Driven Engineering) vision
- **For a better quality: (2 session)**
 - Designs patterns & Refactoring
 - Testing
- **How to reduce risks: (1 session)**
 - Les processes and methods

Grading

- **Evaluation:** The primary basis for evaluation will be 3-4 major projects. 620 students will be required to demonstrate a broader and deeper grasp of subject material in their project submissions. Additional project sections may be required of 620 students.
- **Examinations:** There will be no mid-term examination. There will probably be a final examination.
- **620 Students (only):** Will be required to write a research term paper. The paper will probably have to be presented in class at the end of the semester.
- **Grading:** The student's final course grade will be based primarily upon the student's project grades. The grade on the final examination will carry approximately the same weight as one of the projects. For 620 students, the grade on the research term paper will be weighted at least as heavily as one of the projects. Class attendance, presentations, and participation will also be taken into consideration in determining the final grade in the course.

Lectures

- Software Engineering,
 - Ian Sommerville, Addison Wesley; 8 edition (15 Jun 2006), ISBN-10: 0321313798
 - The Mythical Man-Month
 - Frederick P. Brooks JR., Addison-Wesley, 1995
 - Cours de Software Engineering du Prof. Bertrand Meyer à cette @:
 - <http://se.ethz.ch/teaching/ss2007/252-0204-00/lecture.html>
 - Cours d'Antoine Beugnard à cette @:
 - <http://public.enst-bretagne.fr/~beugnard/>
-
- UML Distilled 3rd édition, a brief guide to the standard object modeling language
 - Martin Fowler, Addison-Wesley Object Technology Series, 2003, ISBN-10: 0321193687
 - UML2 pour les développeurs, cours avec exercices et corrigés
 - Xavier Blanc, Isabelle Mounier et Cédric Besse, Edition Eyrolles, 2006, ISBN-2-212-12029-X
 - UML 2 par la pratique, études de cas et exercices corrigés,
 - Pascal Roques, 6^{ème} édition, Edition Eyrolles, 2008
 - Cours très intéressant du Prof. Jean-Marc Jézéquel à cette @:
 - <http://www.irisa.fr/prive/jezequel/enseignement/PolyUML/poly.pdf>
 - La page de l'OMG dédiée à UML: <http://www.uml.org/>
-
- Design patterns. Catalogue des modèles de conception réutilisables
 - [Richard Helm](#) (Auteur), [Ralph Johnson](#) (Auteur), [John Vlissides](#) (Auteur), [Eric Gamma](#) (Auteur), Vuibert informatique (5 juillet 1999), ISBN-10: 2711786447