



An academic experience:

**The DOSE course
at
ETH Zurich**

Project: involving other universities



Since 2007:

1. ETH Zurich
2. Hanoi University of Technology (Vietnam)
3. ITMO, Russia
4. IT University of Copenhagen, Denmark
5. KAIST (Korea)
6. Odessa National Polytechnic (Ukraine)
7. Politecnico di Milano (Italy)
8. University of Debrecen (Hungary)
9. University of Delhi (India)
10. University of Nizhny Novgorod (Russia)
11. University of Rio Cuarto (Argentina)
12. University of Zurich
13. Universidad Politécnica de Madrid
14. Wuhan University (China)

Project principles and roles



Emulate industrial setting, but only where it makes sense

- Benefits of a controlled setting
- Goal #1 is to learn

All groups created equal

- We do **not** want one university to specify & another implement

Clear management structure

- Central management role, currently at ETH
- Technology choices imposed
 - Eiffel (as a language and method)
 - Origo software development platform
origo.ethz.ch
 - Web tools
 - Any others that may be necessary
- Universities can contribute, e.g. broadcast own lectures

Organization



Cluster-based, not process-based

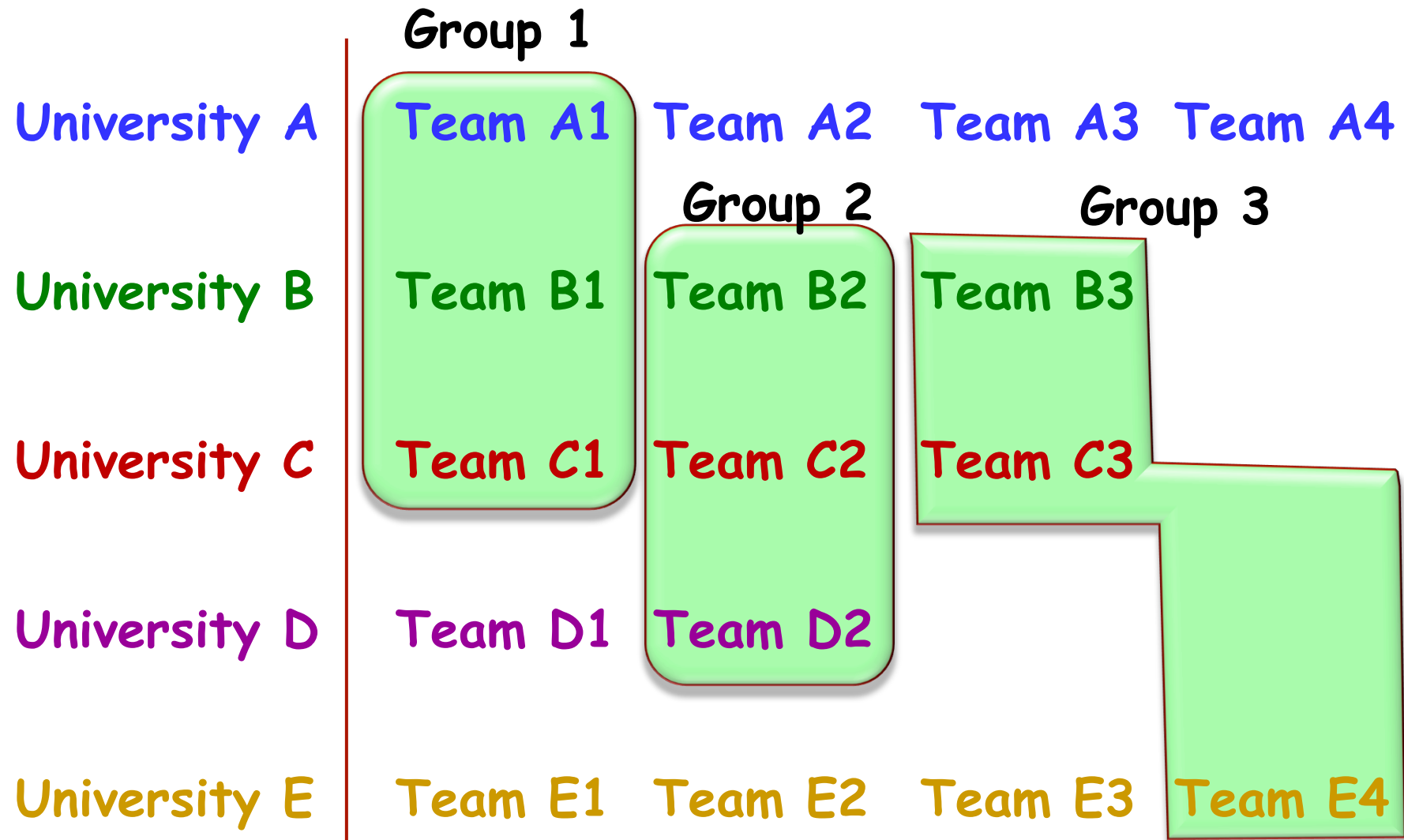
- **Team:** a few students (2 or 3) from one university
- **Group:** collection of teams

Each group does the full project

Each team does a part of the project

This is a part of the system ("Cluster"), not a part of the lifecycle

Teams and groups



Project presentation (2007)



Attended by
students from
all universities
involved

(through Skype)





Universities:

1. ETH Zurich
2. University of Zurich
3. Odessa National Polytechnic (Ukraine)
4. University of Nizhny Novgorod (Russia)

25 developers - 3 countries - 4 projects

Typical group configurations:

- (A) Zurich - (B) Nizhny Novgorod - (C) Zurich
- (A) Zurich - (B) Zurich - (C) Odessa



Phases:

- Phase 1: Requirements (4 weeks)
- Phase 2: Interface specification (3 weeks)
- Phase 3: Implementation & Testing (4 weeks)

Project:

- A system to analyze and store email postings of computer science events (conferences, workshops, etc.), to feed the Computer Science Event List (CSEL, <http://events.informatics-europe.org>)

DOSE 2007 project results



- Delays to set up the projects
- Lack of communication
 - Delay in replying to e-mails
 - Technical problems with Skype conferences
- Misunderstandings in SRS
- Weak API design
 - Incomplete
 - Ambiguous
- Integration partially failed

Software Requirements Specification



D.1. The system shall be able to extract the elements of a call for paper from text e-mails.

D.2. The system can send the e-mail only if at least all key elements have been extracted or introduced by the user. The key elements are: (1) *conference name*, (2) *conference dates*, (3) *abstract and submission deadline*, (4) *conference category*, and (5) *URL of the conference*.

D.3. The conference category is either "Conference" *or* "Symposium" *or* "Workshop" *or* "Summer School"

Some problems



Case 1 - Submission deadline:

- Team A: *day.month.year*
- Team B: integers for the day and year but a string (such as "January" or "February") for the month.

Case 2 - Abstract deadline earlier than submission deadline:

- Team A: Not checked
- Team B: Checked - Exceptions were triggered

More problems



Case 3 - Conference Category:

- Team A: "Conference" *or* "Symposium" *or* "Workshop" *or* "Summer School"
- Team B: "Conference" *xor* "Symposium" *xor* "Workshop" *xor* "Summer School"

Solution: class specification



class EVENT **feature**

submit_to_csel

-- Submit the conference information by sending an e-mail.

require

valid_deadlines: *abstract_deadline.earlier_than (paper_deadline)*

do ... end

feature -- **Implementation**

name: STRING

abstract_deadline, paper_deadline: DATE

category: CATEGORY

invariant

category_status: category.is_conference xor

category.is_symposium xor

category.is_workshop xor

category.is_summer_school

end

Interface: class CATEGORY



```
class CATEGORY feature -- Status report
  is_conference: BOOLEAN
    -- Does this category represent conferences?
  do end
  is_symposium: BOOLEAN
    -- Does this category represent symposiums?
  do end
  is_workshop: BOOLEAN
    -- Does this category represent workshops?
  do end
  is_summer_school: BOOLEAN
    -- Does this category represent summer schools?
  do end
end
```



APIs are critical

Techniques of abstraction & contracts

More problems



Students started assignments too late

Assignments were finished close to the deadline, leaving no time to solve integration problems.

2008 solution:

- Each phase implemented in two cycles
- Students submitted an assignment every second week

Some students dropped out

2008 solution: outside of ETH, choose volunteers; strong pressure on students to commit (or leave early).

Some project could not integrate

2008 solution: force use of contracts

Difficulties (e-mails)



Some members of our team suffer from weak-English.

Aleksey couldn't read any emails last week because his Internet cable had been stolen by a drunk bear.

implemented in EITTEI, now, we cannot integrate it, any hints?

I'm sorry I could not make it to the implementation meeting yesterday. A water pipe in my apartment burst ... After some frantic hours of fixing and cleaning up, it is now more or less OK.

E-mails



... it seems that this team is total absent and reject communication (probably because of a limit in their English)....

Team A complains of a lack of collaboration with its teammates (Team B and C). Also, we received a message concerning the unwillingness of Team B of using X ...

...please fill in the details of your functional requirement...



The last DOSE Courses

1. ETH Zurich
2. University of Zurich **2007**
3. Odessa National Polytechnic (Ukraine)
4. University of Nizhny Novgorod (Russia)
5. Politecnico di Milano (Italy) **2008**
6. University of Debrecen (Hungary)
7. Hanoi University of Technology (Vietnam) **2009**
8. University of Rio Cuarto (Argentina)
9. KAIST (Korea)
10. Wuhan University (China)
11. University of Delhi (India) **2010**



The last DOSE Courses

DOSE 2007

- 25 developers - 3 countries - 4 projects

DOSE 2008

- 42 developers - 5 countries - 21 teams - 7 projects

DOSE 2009:

- 50 developers - 6 countries - 16 teams - 1 project

DOSE 2010:

- 103 developers - 10 countries - 30 teams - 1 project

DOSE 2011:

- 101 developers - 10 countries - 31 teams - 1 project

Workflow



2 weeks

Envisioning

Scope document



3 weeks

Requirements Elicitation

Software requirements specification

2 weeks

Interface Specification

Software requirements Specification + API specified

6 weeks

Development



Groups' presentation



Shenji Schäppi

Computer Science MSc Student at ETH Zurich

- Eiffel Exp.: good
- SRS Exp: good
- Work Exp: Internship at Accenture India (Bangalore)
- O-O languages: Good Knowledge of Java, basic knowledge of C#,C++,C
- Languages spoken: English, German, French



Minh Le Do

Computer Science BSc Student at HUT

- Eiffel Exp.: none
- SRS Exp: basic
- Work Exp: Internship at LINC - HUT (Hanoi, Vietnam)
- Biggest project: 1'000 lines of code
- O-O languages: Basic Knowledge of Java, basic knowledge of C#
- Languages spoken: English, Vietnamese, German



Conrado Plano

Computer Science MSc Student at ETH Zurich

- Eiffel Exp.: good
- SRS Exp: good
- Work Exp: Assistant for lecture Introduction to Programming, Internship at Accenture India (Bangalore) and Lotus Notes Consultant at ATEGRA AG
- O-O languages: Good Knowledge of Java, basic knowledge of C#
- Languages spoken: Spanish, English, German, Italian



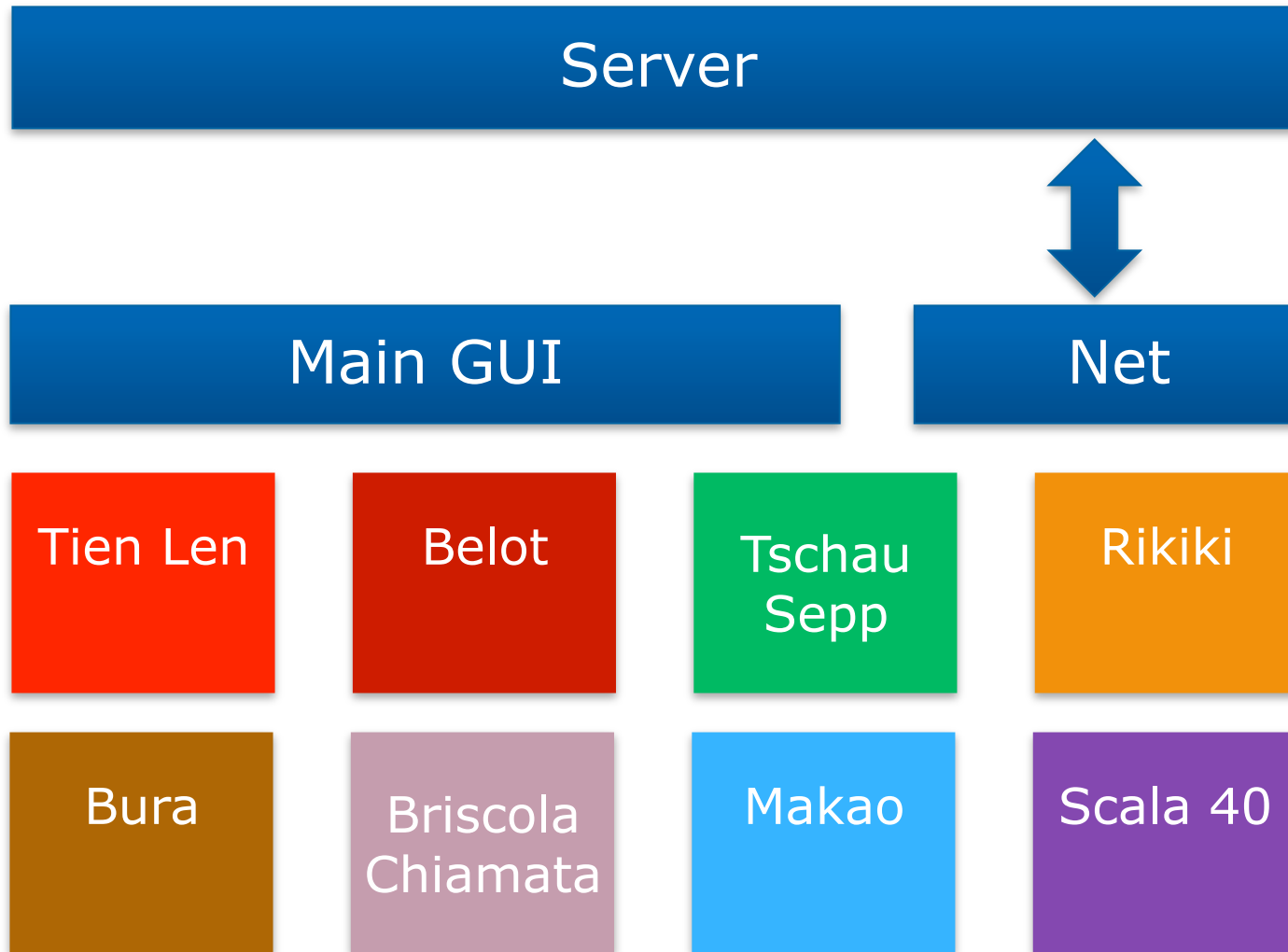
Duc Hoang Bui

Computer Science MSc Student at HUT

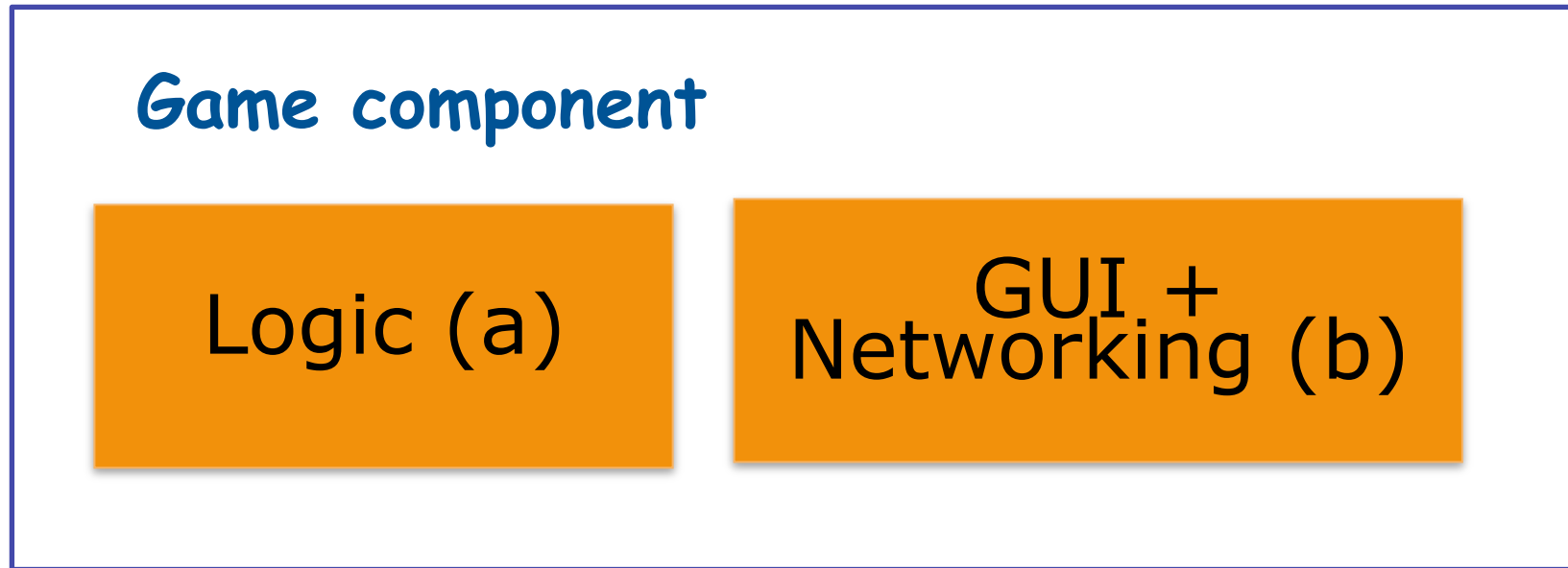
- Eiffel Exp.: basic
- SRS Exp: good
- Work Exp: Internship at ATNAVN (Hanoi)
- Biggest project: 12'000 lines of code(a web application on Struts2)
- O-O languages: Good Knowledge of Java, basic knowledge of C#
- Languages spoken: English, Vietnamese, French



Application Architecture (DOSE 2009)



One game: two teams in two locations



Example: Logic of a Russian game is implemented in by a Russian team
GUI is done by an Italian team.

Organization





The last DOSE Courses: Results

DOSE 2007

- 4 projects failed

DOSE 2008

- 4 projects fully implemented and integrated
- 3 projects failed

DOSE 2009:

- 8 projects fully implemented and integrated

DOSE 2010:

- 11 projects fully implemented and integrated

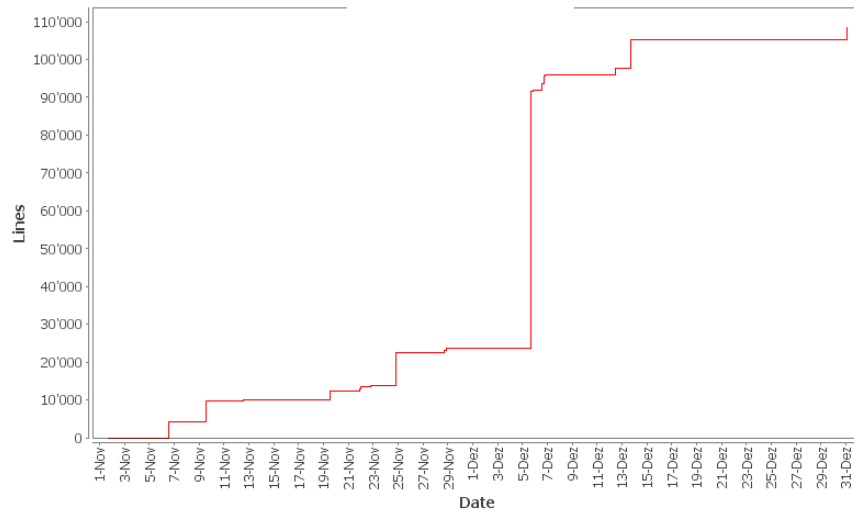
DOSE 2011:

- In progress

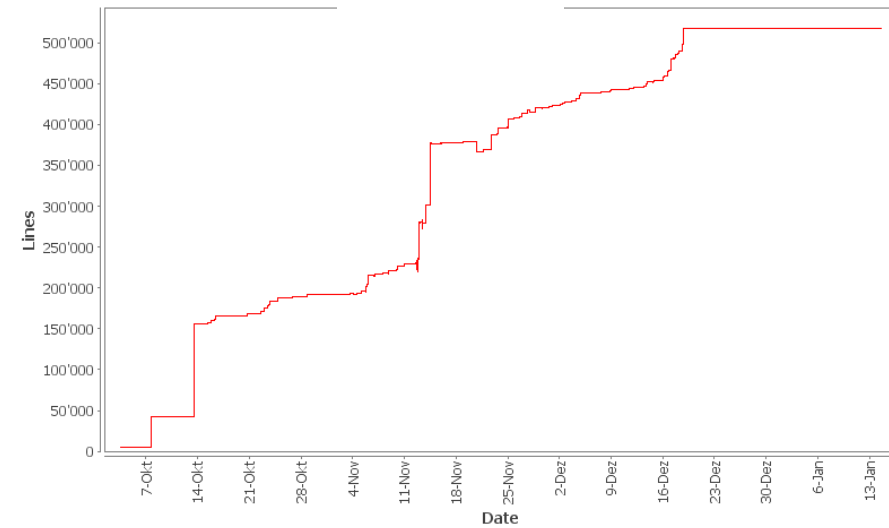
The last DOSE Courses: Results



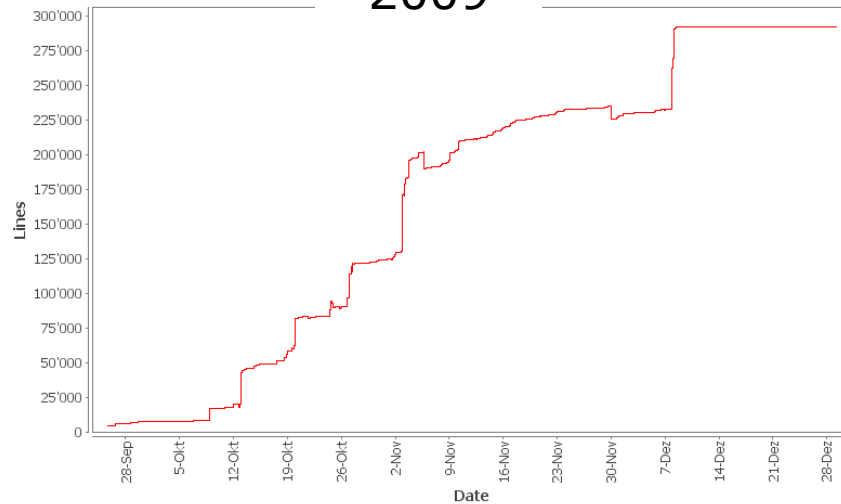
2007



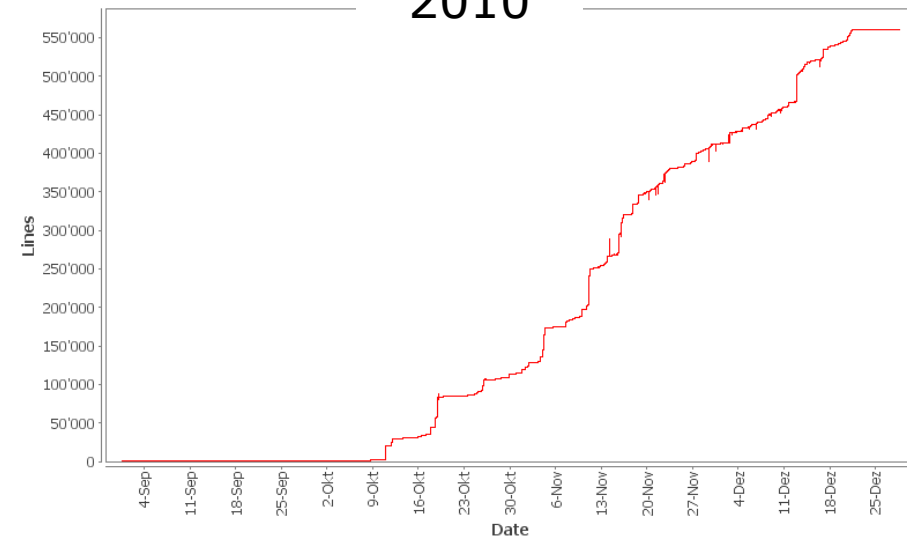
2008



2009



2010



Challenges at DOSE



Project management is difficult

API Design

Communication through phone/skype and video conference is difficult

- Heavy accents
- Noise in communication

Problems at DOSE



...Why is Mitko getting errors while it compiles for me. I think that Mitko might be using an older EiffelStudio...

..There are compilation errors in the code of the teams BriscolaChiamata, Bura, Scala 40, and TschauSepp...

Someone added a cluster but did not update the current project file which produced a broken build

...The GUI works fine in Windows, but it does not work in Linux...

More DOSE problems



There is clear progress in the teams Scala 40 and TschauSepp. But, what is the status of the teams BriscolaChiamata and Bura?

The Vietnamese team promised a new *GUI* by last Monday, but they have not committed yet; what should we do?



Technical problems

Requirements:

- Weak description of functionalities of each component
- Requirements for UI: only high level description, any UI for any game could satisfy the SRS
- Requirements for Logic: the rules of the games are not described
- Requirements for Net: the communication protocol is not specified
- Lack of deep review

Technical problems

API:

- A very small set of functionalities are in the API; main functionalities are missing
- Weak contracts
- Wrong signature for the features
- It is still unclear how implements what

Implementation

- Integration of the components in the last week (instead of integrating them from the first day)
- One team implements most of the project: limited results
- Bad SRS -> delays in implementation

Lessons



Setting up the project is very important: provide templates for documents, basic implementation project, etc

Contracts helps to document and **understand the interfaces**

Critical part of the project should not be outsourced: keep control of what could fail the project

Communication is the core issue

Infrastructure (network, tools...) is critical



DOSE course lessons for the industry:

Challenges and
Practical Advice for
Distributed and Outsourced
Software Engineering

Challenges of DOSE



Project Management

Cultural Differences

Time zones

Communication and Language skills

Challenges: project management



Project management is difficult

In a traditional one-site setting, the manager can just go to a developer's office and ask to see the current state

It is difficult for the project manager to form a good picture of the project's progress

Configuration management plays an important role



Provide templates

Monitor the tasks constantly

Maintain regular communication

For example, one hour weekly meetings

Remind the team about deadlines and double check with the developers if the deadline is still realistic

Practical advice for project management



Require the developers to show a proof of progress (for example by showing a demo, asking deep questions about the implementation, monitoring the code, etc)

Define commit rules

- Code must compile before commit

- Test must run before commit

- Code must review before commit

Apply code reviews:

- Review-to-commit

- Commit-then-review

Challenges: cultural differences



Working in the same culture

Common knowledge

Different cultures

Different cultural backgrounds

Different national holydays

Different interpretations

Cultural differences: train in india



Cultural differences: traffic in Hanoi



Cultural differences: Cambodia





Illustrative Examples (these examples are only illustrative examples)

CULTURAL DIFFERENCES



For example, for Indians:

“yes” means “yes, I have heard you”

“done” means “I will start to do it tomorrow”

Negative feedback is giving by

not responding

Trying not to answer

Suggesting alternatives

[Examples from: Working with India - Wolfgang Messner]



Yes, No: India

<http://www.youtube.com/watch?v=3hCV2oO2akw>

Negative feedback



Company C in Germany sends a feedback form to Company D in India

After 5 days: C contacts D asking about the feedback

Company D: feedback form? When?

[Examples from: Working with India - Wolfgang Messner]

Practical advice: cultural differences



Be aware of the cultural differences and learn about the counterpart's cultures

Indicate the country holidays in a common calendar

Take into account the country holidays when defining a deadline

Practical advice: time zones



Keep meetings on schedule

Keep in mind the Daylight Saving Time

Do not wait to send an e-mail (even if the it is late or early in the other time zones)

Challenges: communication and language skills



E-mail is not enough - need for voice communication

Communication through phone/skype and video conference is difficult

Heavy accents

Different English mistakes to the ones one is used to

Tools are important

More practical advice



Use several forms of communications: e-mail, voice conferences, wikis, docs

Create mailing lists

Send the important information in writing

Write minutes of the meetings recording decisions taken, and action items (todos)



**The world has gone global,
so has the software world**





**Many issues remain,
failure always possible**





**Solutions exist,
improving all the time**





**Many software engineering lessons
apply, made even more relevant by
distributed development**





Communication is the core issue:

- **Between people**
- **Between modules: crucial role of APIs and contracts**





Infrastructure is critical





**Technology changes our
mode of working**





**The written word
remains essential**





**We can't do this without
O-O and contracts**





Universities should teach this

