

Projektityö-kursseista...

TTY/OHJ:

www.cs.tut.fi/kurssit/OHJ-3500

TaY/TKT:

www.cs.uta.fi/pt

*Tämä esitys kertoo Tampereen yliopiston
Tietojenkäsittelytieteiden laitoksen ja Tampereen
teknillisen yliopiston Ohjelmistotekniikan laitoksen
opiskelijaryhmien ohjelmistoprojektikursseista.*

TYÖAIHEITA SAA EHDOTELLA.

Teaching Software Projects at Universities in Tampere

**Lecturer Tero Ahtee, TUT, Institute
of Software Systems**

**Lecturer Timo Poranen, UTA,
Department of Computer Sciences**

TUT: 8 prof., 80 staff. Annually 100 masters, 2 PhD.

UTA: 8 prof., 90 staff. Annually 50 masters, 5 PhDs.

Motivation

- **Software project courses are important** for all information technology students:
 - The course integrates the knowledge and skills from earlier courses
 - (Probably) All software products are constructed in industry in "projects" or projects. Project skills are essential for all students
 - The course gives a possibility to learn how to make a large software in a right way
- Provides a natural way to co-operate with industry (and other universities etc.)

Contents of this presentation

- Software projects at Tampere University of Technology and at University of Tampere
- Educational aims
- Main quality issues
- Lessons learnt in teaching
- Success and failure factors for projects
- Course improvement ideas
- Conclusions, and questions

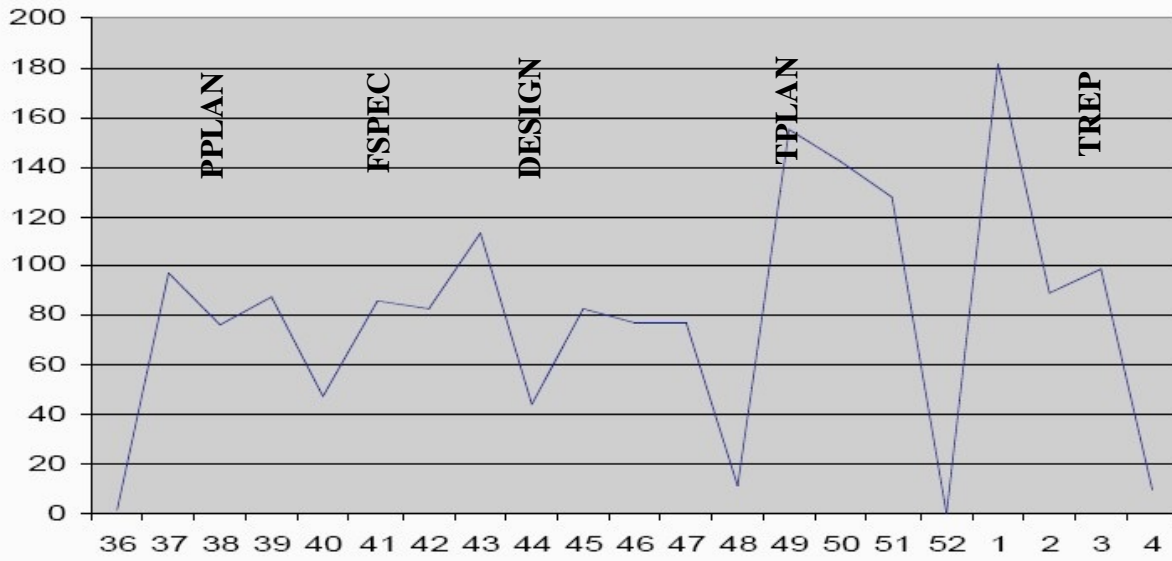
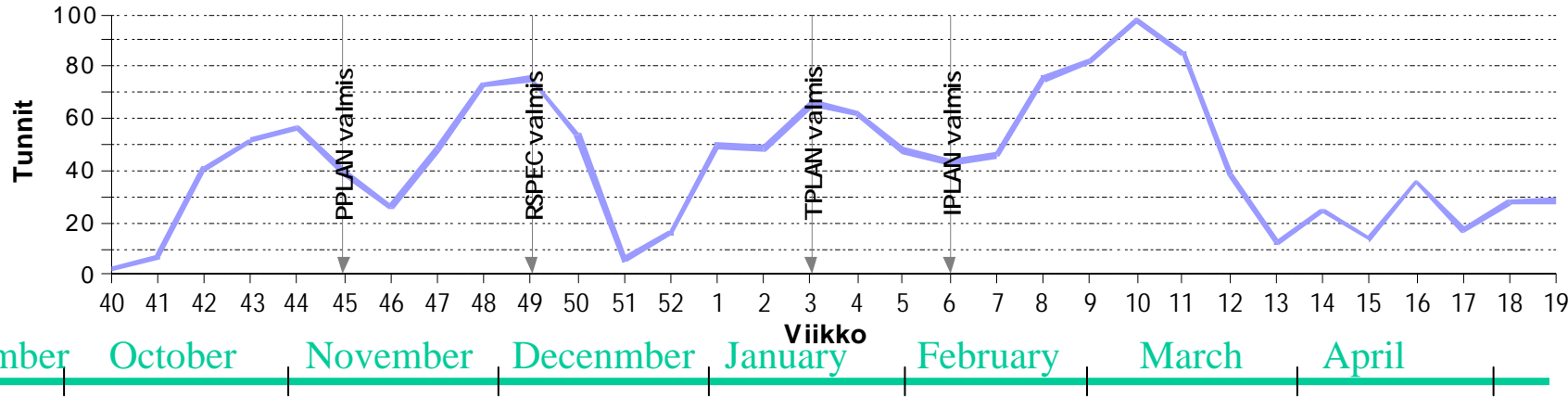
Project courses' overview

| | UTA (3rd year) | TUT (4th year) |
|-------------------------------|--------------------|-------------------|
| Group forming | Set, by staff | Free, by students |
| Project manager | From senior course | From group |
| Project subject | Set, by staff | Free, by students |
| Lecture hours | 16 | 34 |
| Duration (weeks) | 30 | 20 |
| Staff (full + part-time) | 1+0 | 1+9 |
| Possible for foreign students | Yes | Yes |

Project subjects 2006-07

- TUT
 - 12 www
 - 4 mobile/embedded
 - 10 stand-alone applications.
- UTA:
 - 13 www
 - 3 mobile
 - 2 stand-alone applications.

Sample schedules



Project Course statistics 1

Table 3: Size measurements of the Project Work course at TUT and UTA.

| Academic year | Project size at TUT (avg. hours) | Course size at TUT (groups) | | Project size at UTA (avg. hours) | Course size at UTA (groups) |
|---------------|----------------------------------|-----------------------------|--|----------------------------------|-----------------------------|
| 1997-98 | 937 | 30 | | - | - |
| 1998-99 | 1248 | 17 | | - | - |
| 1999-2000 | 1073 | 27 | | - | - |
| 2000-01 | 1054 | 26 | | - | - |
| 2001-02 | 1152 | 25 | | - | - |
| 2002-03 | 1053 | 24 | | - | 14 |
| 2003-04 | 1194 | 25 | | - | 15 |
| 2004-05 | 1256 | 23 | | - | 13 |
| 2005-06 | 1239 | 21 | | 1008 | 19+1 |
| 2006-07 | 1314 | 26 | | 1089 | 18+2 |

Project Course statistics 2

Sample data from TUT projects 2006-07:

- 2946 - 34304 LOC
- 2574 - 27839 SLOC
- 18 - 160 classes
- 5 - 20 screens
- Up to 213 code files
- Productivity: e.g. 64 LOC/day
- Productivity: e.g. 10 LOC/hour (total hours)
- 285 - 709 document pages.

Course goals and educational aims

- Working in a large (1000 hours) sw project
- From requirements via design to features
- Constructing a real software product
- Writing technical documentation, understanding documentation
- Inspections (the formal ones, not reviews)
- Communication is not easy
- Group working
- Presentations
- Number of working days != calendar days (Xmas...)
- Students have to think (a lot)!
 - There is no silver bullet!

Quality issues 1: documents

- Both courses have extensive document templates
- Compulsory documents:
 - Preliminary analysis / Feasibility study
 - Project plan
 - Requirements specification
 - Design document
 - Test plan
 - Test report
 - Final report
 - User manual, maintenance guide (at TUT)
- Other docs: depends on the project!

Quality issues 2: life-cycle models

- At TUT the used life-cycle model has not been set to any specific. Most have followed more or less waterfall model. Last course had 5 incremental groups. Early prototype has been encouraged.
- At UTA, students can choose their model, too. This started last year, before that only waterfall model was allowed.
- For UTA development model distribution, see page 2 of technical report "Software Projects 2007"
 - <http://www.cs.uta.fi/reports/dsarja/D-2007-7.pdf>
 - 8 Incremental, 6 waterfall, 4 iterative.

Quality issues 3: working methods

- Weekly reporting is compulsory
- Groups are encouraged to meet weekly, “internet meetings” should be used only in exceptional cases. Discussion forums are ok!
- (Programming) Workshops (hack fest / code camp) are often productive
- Project organisation at UTA: Project Manager and team. Steering group contains at least client, project manager, and lecturer. Meetings only in emergency situation.
- Version management (CVS, Subversion)
- Compulsory inspections with some regular meetings with the course staff.

Quality issues 4: weekly reporting

PROJECT _XYZ_ WEEKLY REPORT

Subject: Project XYZ week 42

PROJECT: (name or identification)

DATE, NUMBER OF WEEK:

IN WHAT PHASE OF PROJECT WE ARE NOW:

SCHEDULE (OK or changes if exist):

WHAT HAVE BEEN DONE AFTER PREVIOUS WEEKLY REPORT:

WHAT WILL BE DONE BEFORE NEXT WEEKLY REPORT:

NEXT MILESTONE/DELIVERABLE (what and when):

WORKING HOURS OF PROJECT PERSONNEL:

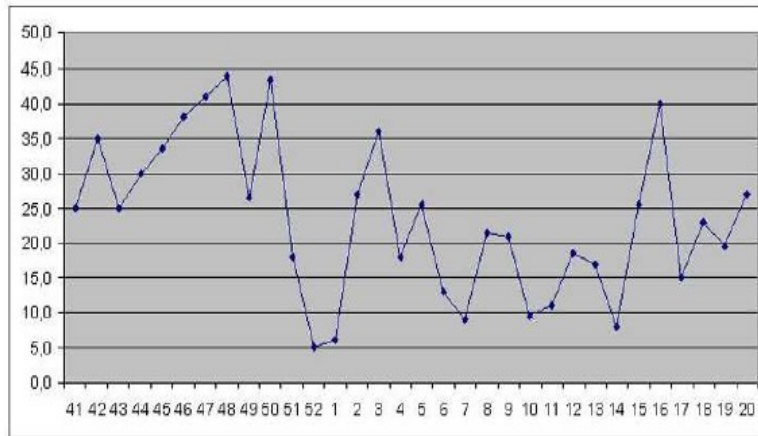
PROBLEMS IF EXIST (why ? how to solve them ?):

WHAT PLANS HAVE BEEN CHANGED:

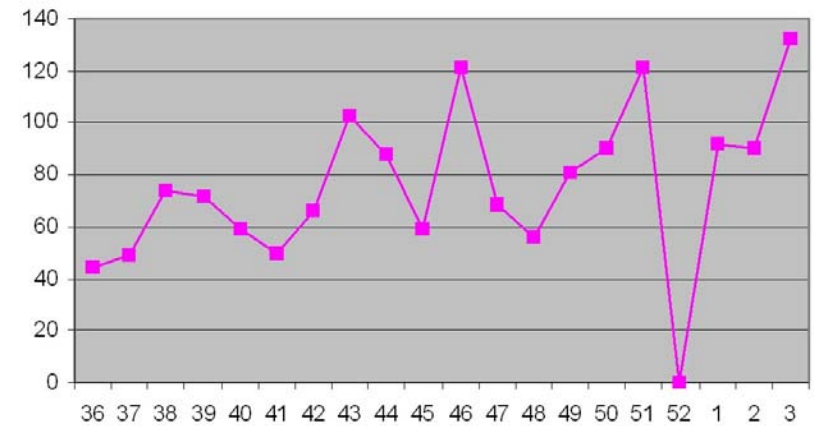
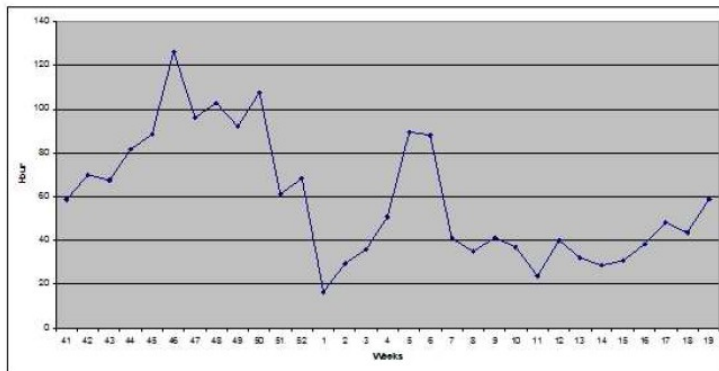
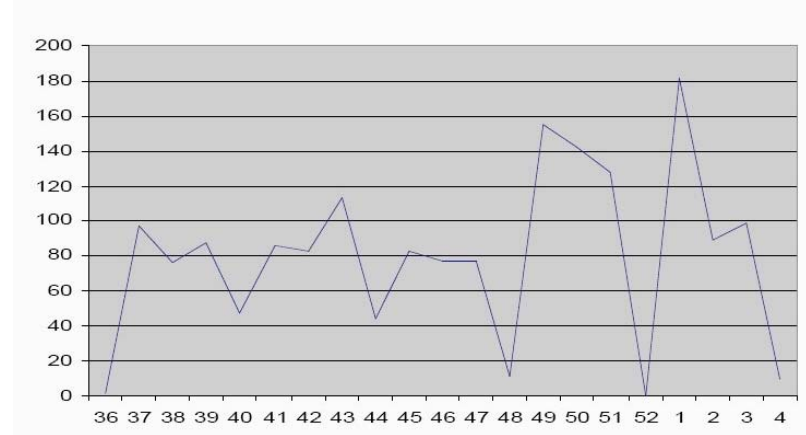
MISCELLANEOUS (OTHER THINGS TO MENTION):

Quality issues 5: work distribution

Some UTA weekly working hours



Some TUT weekly working hours



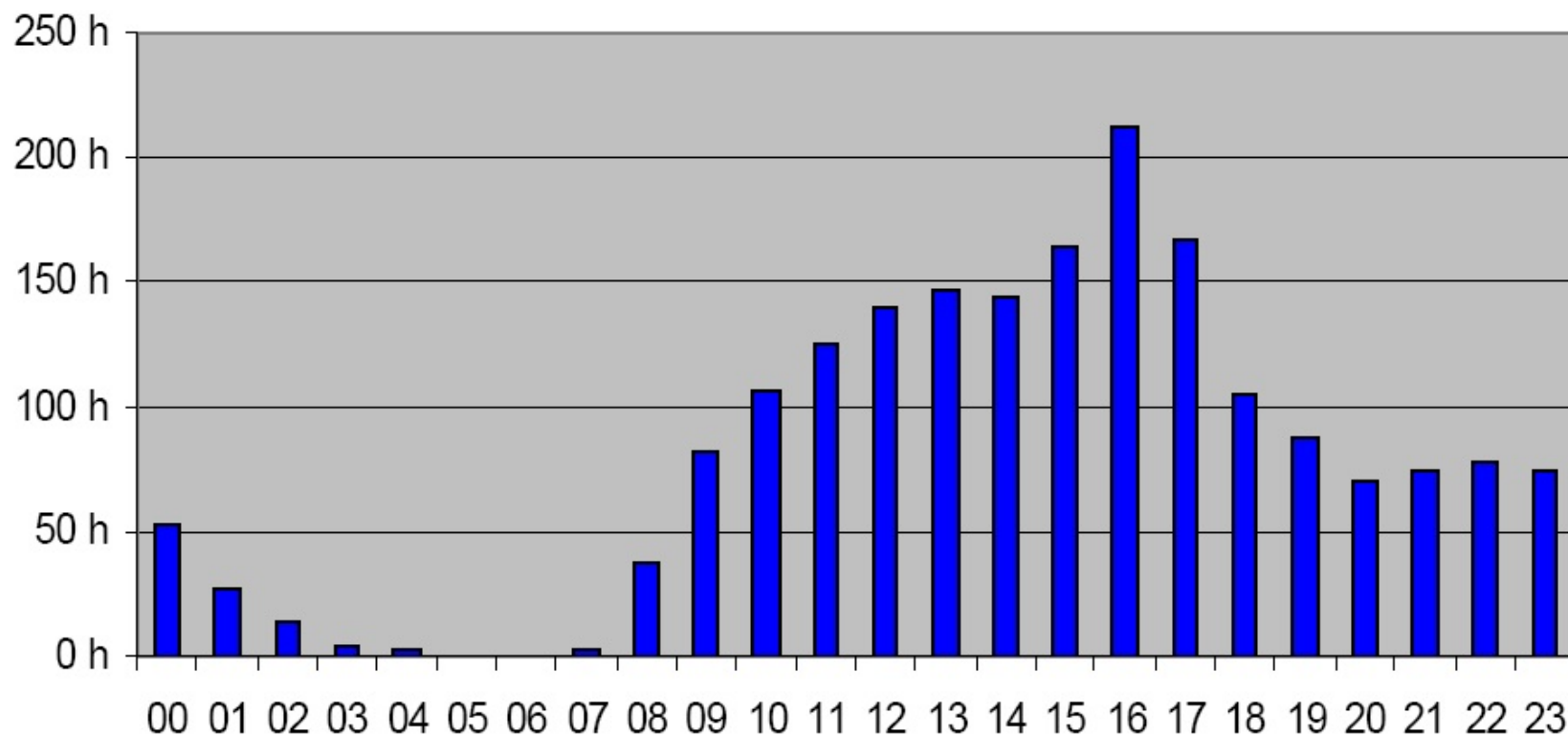
Quality issues 6: Working hour table

An example of TUT working hours table

| | FeasStu | FuncSpec | Design | Implem | Testing | Ins, U, Mai | Other | Total | % |
|----------------|--------------|------------|---------------|---------------|---------------|-------------|------------|---------------|-----|
| Tumi | 10:30 | 8 | 13:45 | 17:30 | 10:30 | | 13 | 73:15 | 6% |
| Meet | 28:45 | 12:30 | 15 | 15:30 | 18:15 | | 18:30 | 108:30 | 9% |
| Insp | 7 | 5:30 | 18:30 | 15 | 6:30 | 4 | 2:30 | 59 | 5% |
| Study | 9:30 | 2 | 16 | 49 | 15 | | 15 | 106:30 | 9% |
| Docum | 32 | 80:30 | 59 | 41:30 | 30 | 1 | 46:15 | 290:15 | 25% |
| Pro/D | | 17 | 4 | 8 | | | | 29 | 2% |
| PrjMgmt | 4 | 4 | 3:30 | 12:30 | | | 19:30 | 43:30 | 4% |
| Work | | 12:30 | 10:30 | 311:15 | 40:30 | 22 | 73:15 | 470 | 40% |
| | 91:45 | 142 | 140:15 | 470:15 | 120:45 | 27 | 188 | 1180 | |
| % | 8% | 12% | 12% | 40% | 10% | 2% | 16% | | |

Reported daily working time, according to CVS statistics (TUT)

Työskentelyajankohta



Quality issues 7: Inspections

Needed information for statistics:

- preparation time
- inspection time
- number of findings
- number of pages.

TUT: 5 + 1

UTA: 4 inspections.

INSPECTION DIARY (SUMMARY) GROUP: _____
 PROJECT: _____ TARGET OF INSPECTION: _____
 INSPECTION DATE: _____ TIME: _____ - _____

| INSPECTION TEAM NAME | TIME USED (hours) | | | |
|-------------------------|-------------------|---------|-------|------|
| | PREP | INSPECT | TOTAL | CORR |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

| RESULT OF INSPECTION | |
|------------------------|-----|
| SUMMARY OF FINDINGS | |
| SEVERITY | PCS |
| Fatal | |
| Medium | |
| K Cosmetic | |
| requires Study | |
| P missing | |
| L too much, not needed | |
| SOURCE | PCS |
| MÁ requirements | |
| SU design | |
| TO implementation | |
| MU other | |

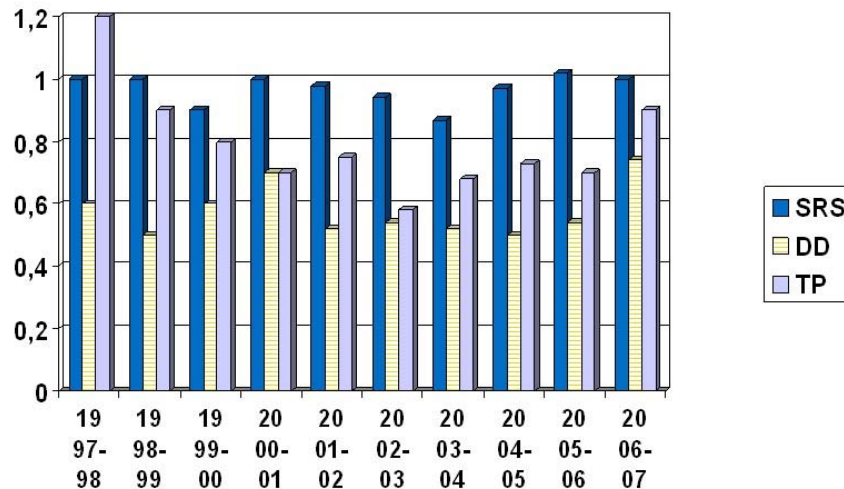
Total amount of findings pcs

Pages in document:

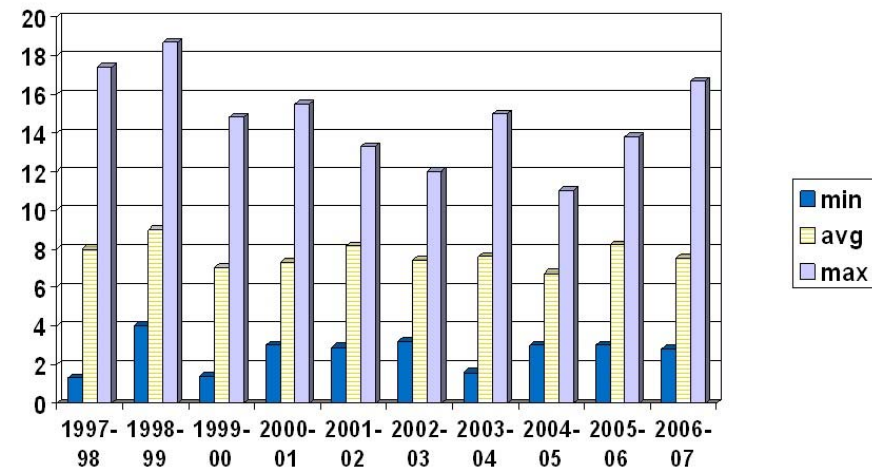
| RESULT OF INSPECTION: | date | signature |
|----------------------------|--------------------------|-----------|
| ACCEPTED AS IS | <input type="checkbox"/> | _____ |
| ACCEPTED AFTER CORRECTIONS | <input type="checkbox"/> | _____ |
| NEEDS ANOTHER INSPECTION | <input type="checkbox"/> | _____ |
| OTHER | <input type="checkbox"/> | _____ |

Quality issues 8: Inspection statistics

Document inspections at TUT
findings / page (average) (5/7)



Document inspections at TUT,
% of project time spent to inspections (7/7)



Students like very much all kind of feedback.

Quality issues 9: final report, presentation and meeting

- After the project, groups have a public final presentation. Groups write final report, according to a document template.
- Main issues are
 - Risks and risk management during the project
 - Project management
 - Methods and tools (e.g. working hours)
 - Experiences and lessons learnt
 - Comments about the course
 - Statistics
- Final report is discussed with the group in the final meeting.

Lessons learnt (teaching)

- Try to collect data from projects systematically
- There is no upper limit for supervision hours for some project groups...
- A reasonable "guidance limit" and the number of course staff is a compromise (dept's resources)
- Co-operation with other courses (and universities) takes time, but is often very fruitful and gives realism to the projects
- All projects are different, so you won't get too bored... (adds motivation in many ways)
- How to combine teaching and research? Own research projects and then publications...

(a university lecturer is not required to do any research or PhD studies)

Project success factors

- Motivated students (almost all are)
- A client who has time to answer group's questions, and who is motivated, too
- Already familiar with tools
- "Cool" and interesting project topic
- Not too large and heavy project size
- Problems should be handled when they occur, not after the project!
- Programming should be started quite soon to avoid too optimistic productivity estimations. Especially in "new technology" projects.

Project failure factors

- **Working while studying (most students do)**
- Poor understanding about project's subject and client's business
- New tools
- No experience in estimating project size
- Third party components

- "Trust me, this is a small and easy project... a trained monkey can code this in three months"
- "I think you already know that, so I don't need to mention or document it... it is so obvious that everybody knows it."

Future development ideas

- More collaboration with other courses (e.g. testing)
- More incremental/agile projects
- Comparison and study of groups' working hours between universities (statistics).

There is still a major challenge: students do not learn project work and management, and project size estimation, by reading a book. They have to complete a real project to learn those skills, and that takes time.

Conclusions

- Can, and should, be department's "flagship" course
- Gathers student's all previous knowledge and skills
- Needs good basis; several large programming courses, sw eng basics and methods, data structures, sw architecture, testing, usability, databases, and maintenance courses
- Is demanding groupwork, but rewarding
- Respected by industry companies
- Enables seeding good sw eng practices to industry (e.g. inspections and final reports)
- Students consider Project Courses very good and important for their future.

Some early hours' work during cold dark nights...

Test plan: "8.3.1.4 Test Group: Exit The Program

There may be some impossible ways to exit the program that are not tested."

Design document: "2.1 Application environment

It is an independent single-user software that doesn't need any other programs but by itself it doesn't do anything."

Project plan: "3.1.1 To the client

When the project completes with success, the product is to be a part of later-finishing, more exhausting application."

[should be "exhaustive"]

We would like to hear other projects' experiences and statistics !

tensu@cs.tut.fi
tp@cs.uta.fi