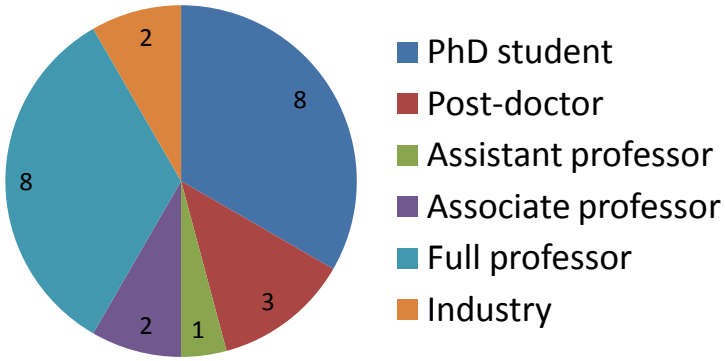
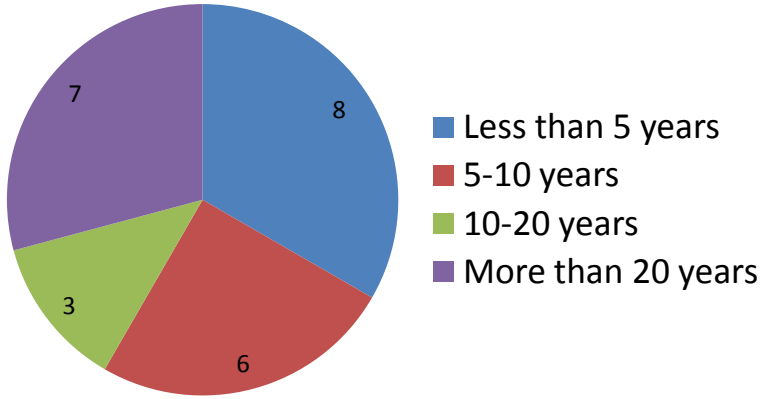
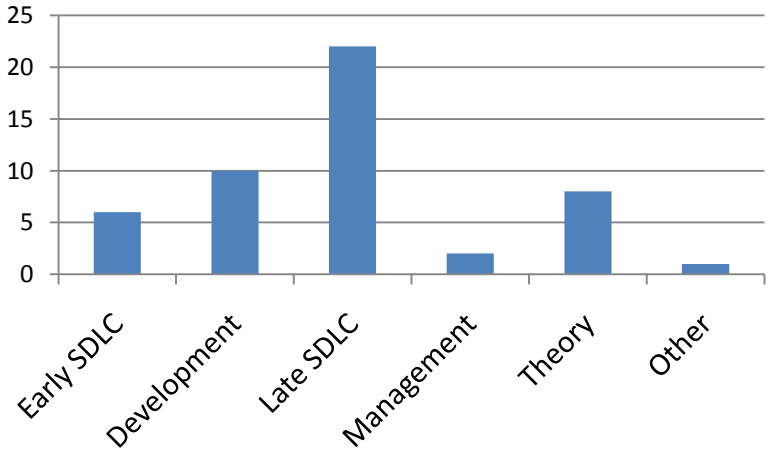
Evolution aware Software Testing and Debugging

Mentor:
Abhik Roychoudhury
Leader:
Tamás Gergely
Members:
Casandra Holotescu
Pooyan Jamshidi
Salman Mirghasemi
György Orbán
Hoang Duong Thien Nguyen

General Information

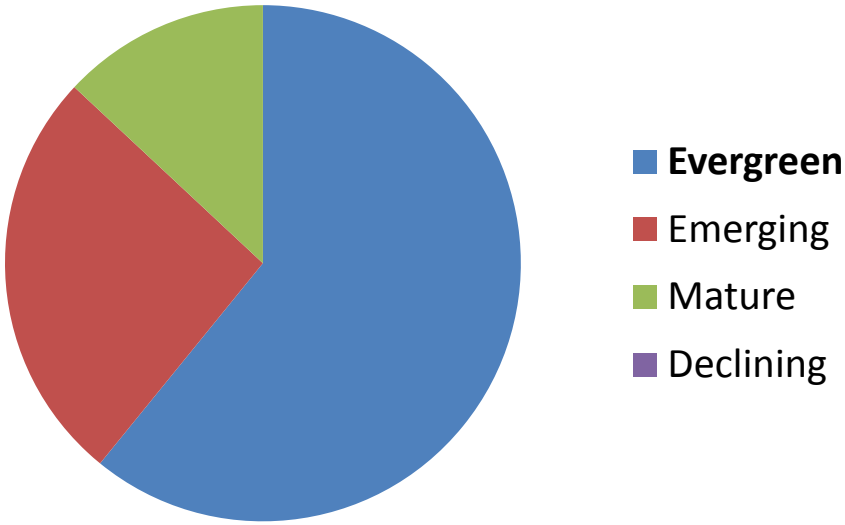


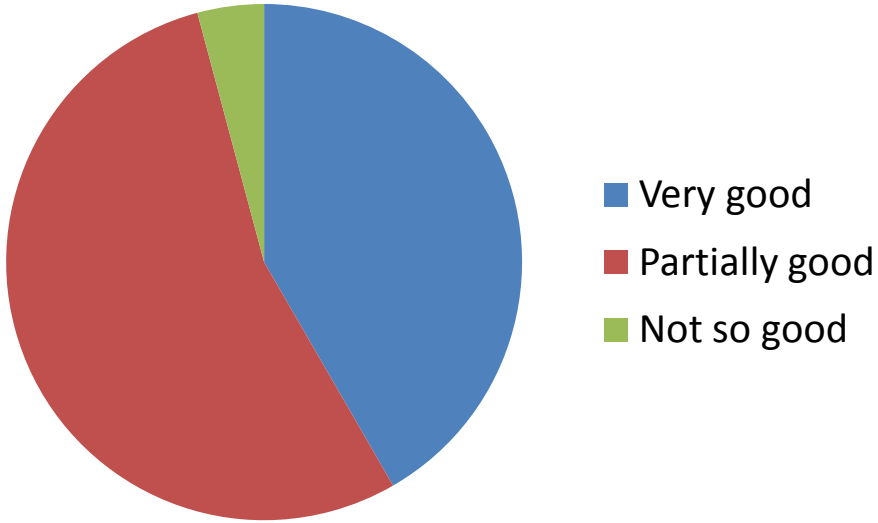




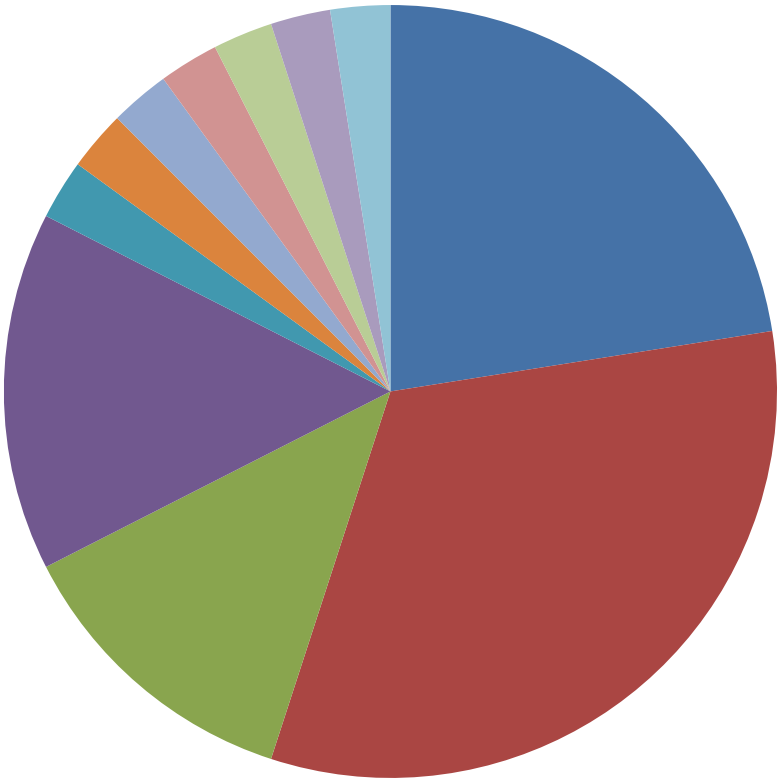
- Data Items: 1244
- Number of interviewees: 25

Maturity of the topic and quality of the questions



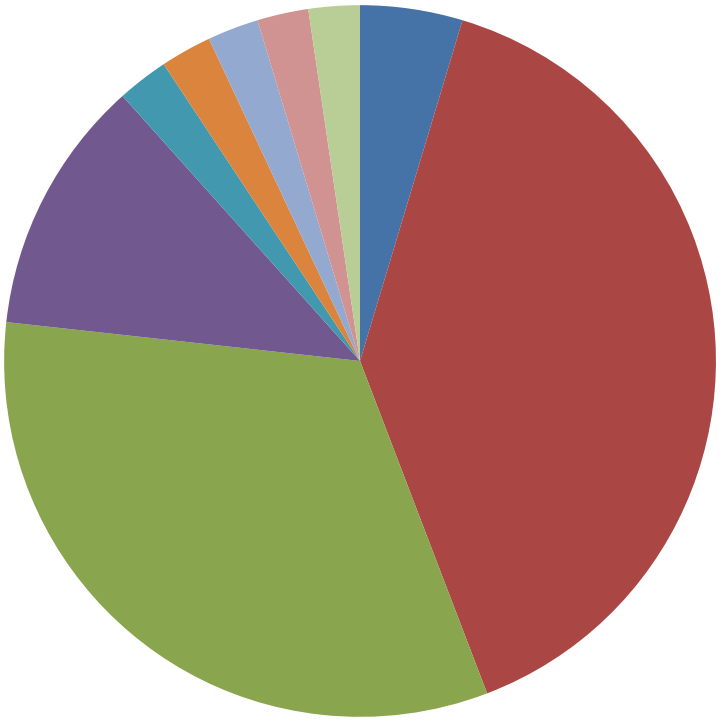


Common Error Types



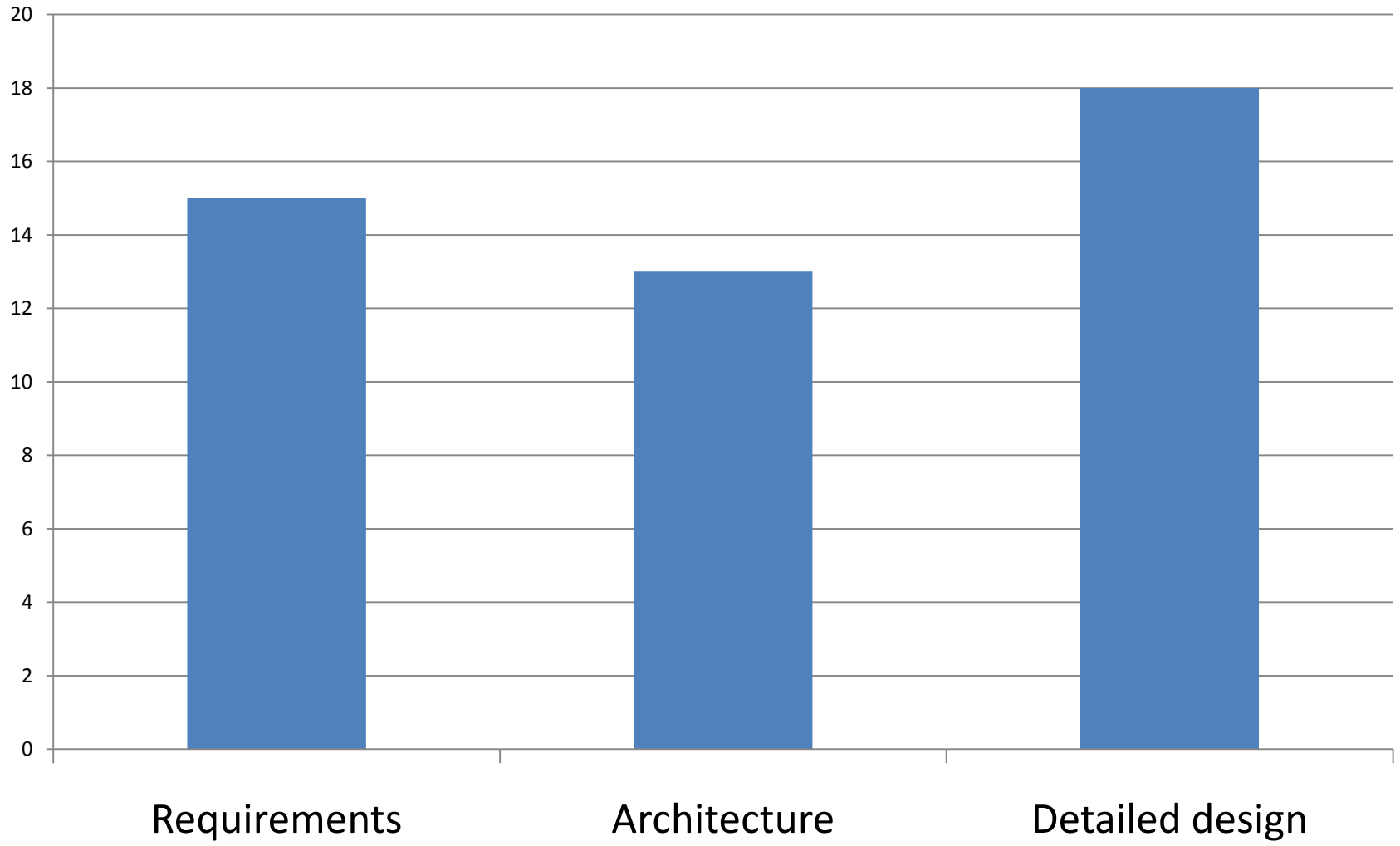
- **Branch condition errors or assignment statement errors**
- **Wrong usage of interfaces**
- Wrong constant value usage/handling
- **Language specific errors**
- You can break old functionality or incorrectly implement new functionality
- Documentation error
- Incorrect/partial bugfix
- Inconsistent data usage (as constatn/value handling)
- Errors on borders between components (as wrong interface usage)
- concurrency error (deadlock, data-race introduced due to changes)
- some errors are introduced by copy and paste

Ways of discovering/preventing regression errors



- Coverage based testing is enough in most cases
- **Specific change-stressing techniques for test-suite augmentation are needed**
- **Re-execution of functional tests discover the important errors**
- **Specific programming environment can help preventing these bugs**
- Code review
- Automated testing
- Change-based retesting
- Static and dynamic analysis tools
- Command line

Possible levels of regression testing (apart from code level)



Some correlation analysis

	Q1.a	Q1.b	Q1.c	Q1.d	Q1.e
Q2.a	-0,22	-0,01	-0,15	-0,17	-0,20
Q2.b	-0,38	0,20	0,34	0,18	-0,08
Q2.c	-0,01	0,12	0,24	0,31	0,26
Q2.d	0,04	-0,32	-0,25	-0,28	-0,13
Q2.e	0,25	0,08	0,25	0,42	0,51

1,00	-0,06	0,29	-0,20	0,13
-0,06	1,00	0,27	0,34	0,22
0,29	0,27	1,00	0,41	0,35
-0,20	0,34	0,41	1,00	0,48
0,13	0,22	0,35	0,48	1,00

1,00	0,19	0,25	0,22	-0,13
0,19	1,00	0,02	-0,35	-0,20
0,25	0,02	1,00	0,23	0,15
0,22	-0,35	0,23	1,00	-0,23
-0,13	-0,20	0,15	-0,23	1,00

Some interesting insights

- More experienced people thinks that test driven development is a good method to avoid / detect regression errors, while overall, TDD is considered the least appropriate method for it.
- People do not think that
  - coverage-based techniques are enough to detect regression errors (only 8%)
  - an IDE used by developers can help much on regression testing (20%)
- Continuous integration and Pair programming are told to be important to avoid regression errors

An interesting but not surprising outcome

- People from Academy trust in some certain techniques/practices

- Industry people know that

"it depends on ..."