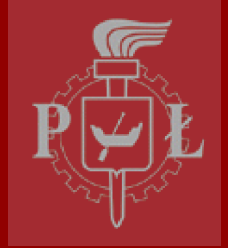
Software Engineering in Development of Software for LLRF System

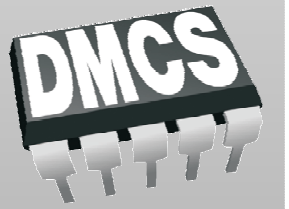
A. Piotrowski

Technical University of Lodz, Poland
Department of Microelectronics and Computer Science

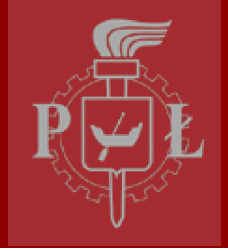
December 12, 2011



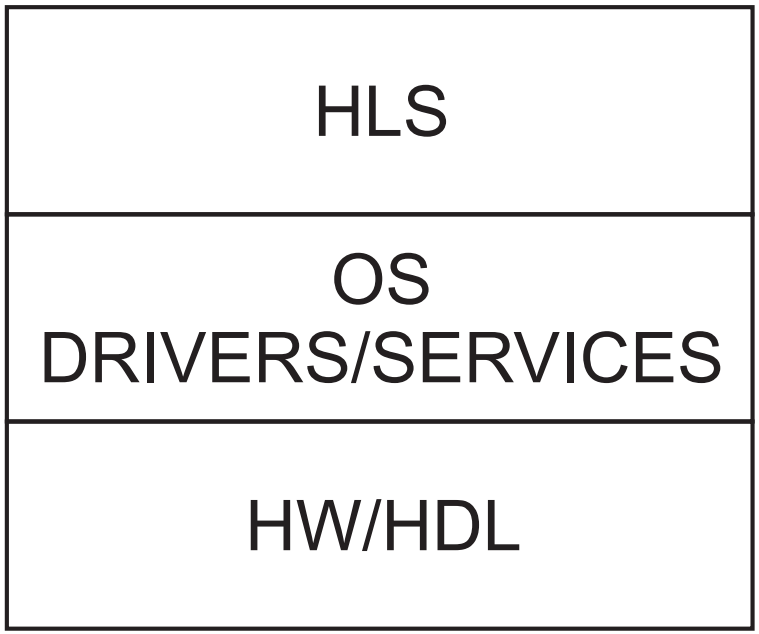
Agenda

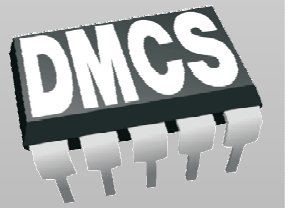


Technical University of Lodz, Poland
Department of Microelectronics and Computer Science

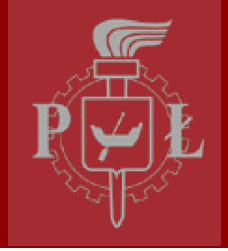


Software Layers



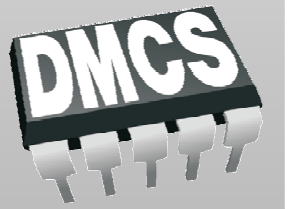


Technical University of Lodz, Poland
Department of Microelectronics and Computer Science

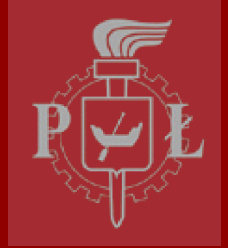


Common Problems with Software Development

- Delivering New Features to Customer Takes Too Long
- Quality Delivered to Customer Is Unacceptable
- Features Are Not Used by Customer
- Software Is Not Useful to Customer

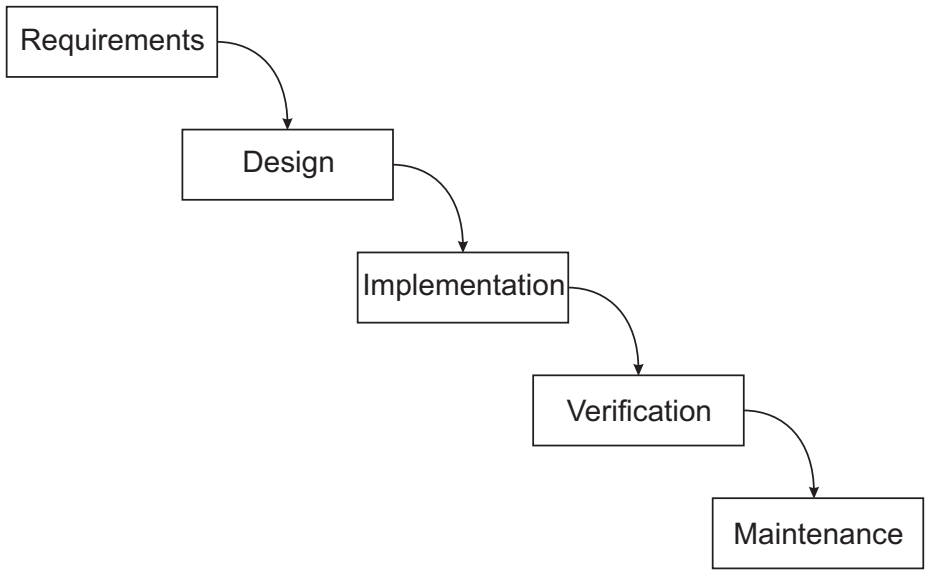


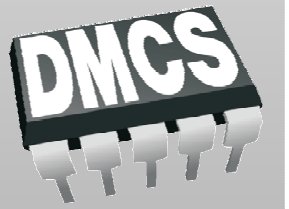
Technical University of Lodz, Poland
Department of Microelectronics and Computer Science



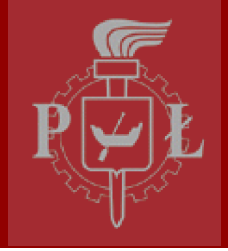
Software Engineering

- The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software
- Software development methodology:
  - Waterfall development
  - Iterative development e.g. agile software development



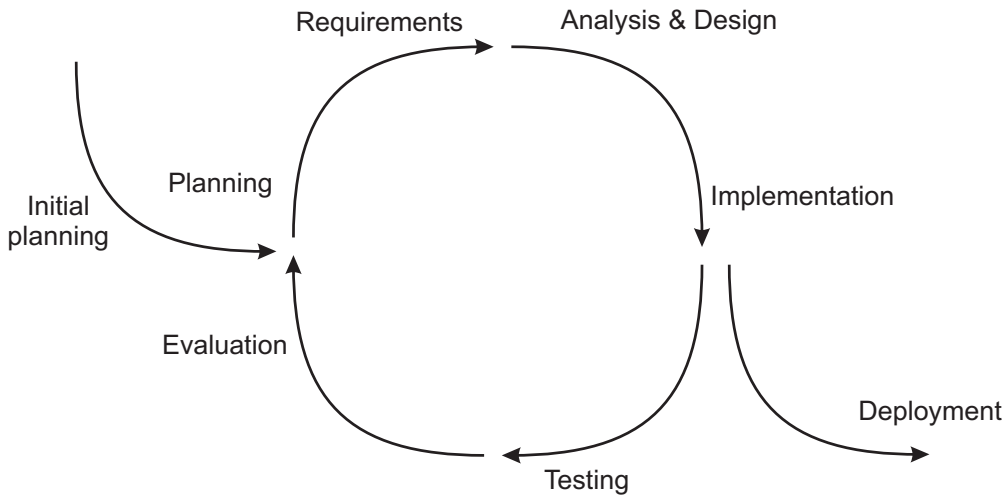


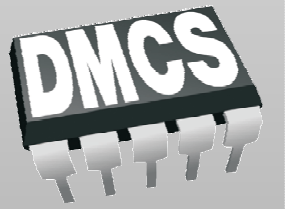
Technical University of Lodz, Poland
Department of Microelectronics and Computer Science



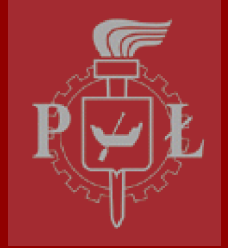
Software Engineering

- The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software
- Software development methodology:
  - Waterfall development
  - Iterative development e.g. agile software development





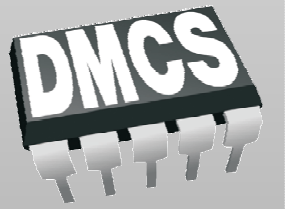
Technical University of Lodz, Poland
Department of Microelectronics and Computer Science

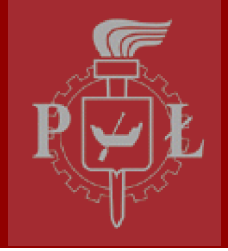


Commercial Projects vs Software Development for High Energy Physic Experiments

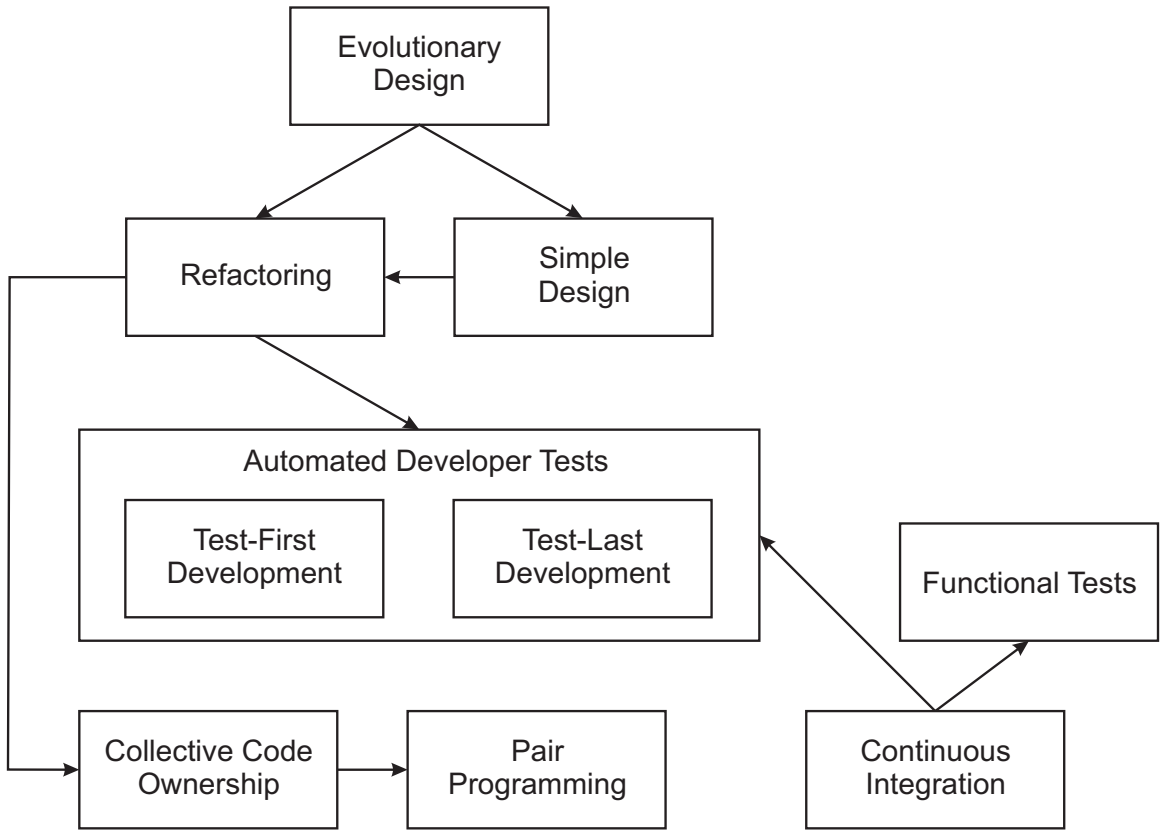
- team composed of several persons
- requirements can be changed but in the last stage of development are fixed
- small influence of environment on developed system

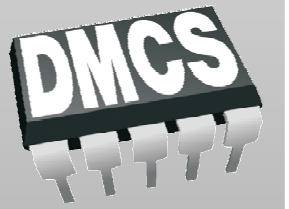
- team composed of one or two person
- requirements can be changed all the time
- big influence of environment on developed system



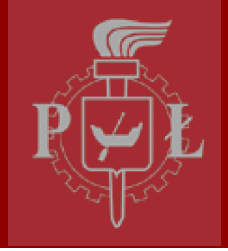


Flexibility Practices



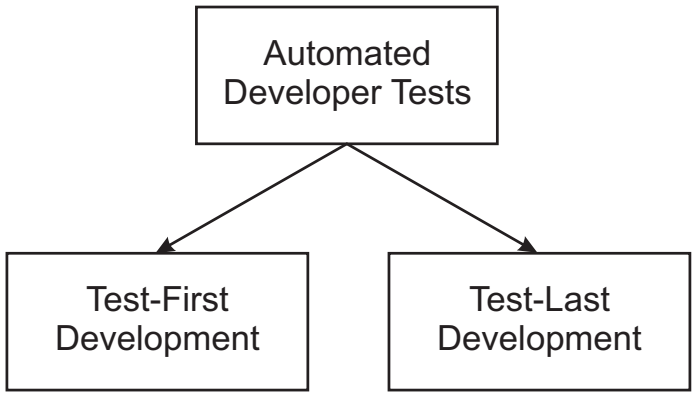


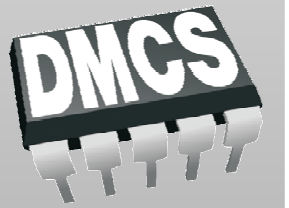
Technical University of Lodz, Poland
Department of Microelectronics and Computer Science



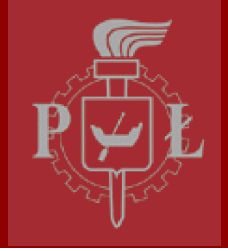
Automated Developer Tests

The Automated Developer Tests pattern is a set of tests that are written and maintained by developers to reduce the cost of finding and fixing bugs - thereby improving code quality and to enable the change of the design as requirements are addressed incrementally.
The Automated Developer Tests pattern improve project testability.



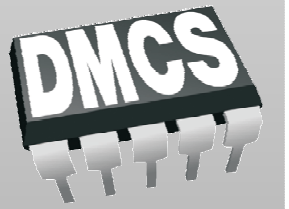


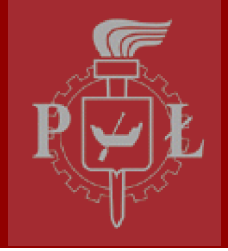
Technical University of Lodz, Poland
Department of Microelectronics and Computer Science



Test-Last Development

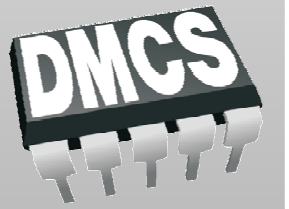
- The practice of Test-Last Development involves writing tests after writing the code to support the requirements for a particular task.
- Tests exercise the system after it has been built.



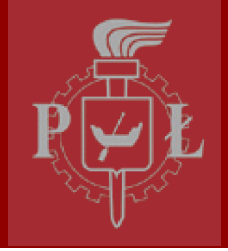


Test-First Development

- The Test-First Development practice involves writing tests before writing the production code that will support and eventually pass that test.
- Tests resulting from this practice tend to reflect a developers understanding of requirements because there is no design at its inception.



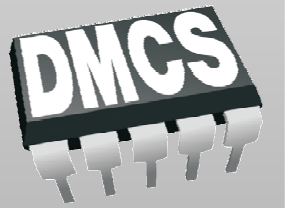
Technical University of Lodz, Poland
Department of Microelectronics and Computer Science



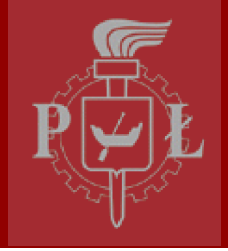
Refactoring

Some problems are a natural result of software development.

- Software gains entropy over time.
- Quick fixes quickly build up a design makes code more difficult to understand and modify.
- Requirements are added and modified, and the current design is no longer a good solution to the problem.
- Code duplication is almost inevitable to avoid changing working code and possibly introducing a bug.
- Software development is a learning process, therefore design decisions with todays knowledge can be better adjusted.

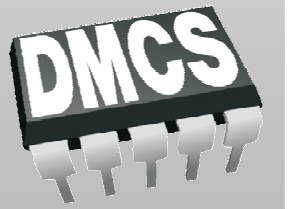


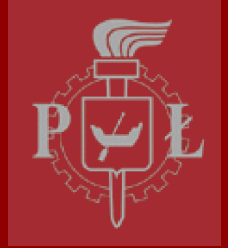
Technical University of Lodz, Poland
Department of Microelectronics and Computer Science



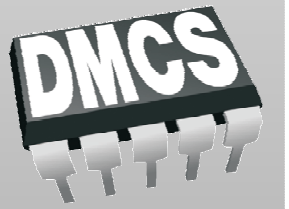
Summary

- Software engineering methodology can be used to increase software reliability and limit amount of time required for bug fixing.
- Software engineering is utilized with success with commercial software projects why do not use this approach in HLP experiments.





Thank You



Technical University of Lodz, Poland
Department of Microelectronics and Computer Science