Using open source paradigms to teach system development

Dimitrios Platis
@PlatisSolutions
dimitris@platis.solutions

About me

Dimitrios Platis

- Grew up in Rodos, Greece
- Software Engineer @ Zenseact, Gothenburg
- Course responsible @ DIT112, Gothenburg University
- Interests:
  - Embedded systems
  - Software Architecture, API Design
  - Open source software & hardware
  - Robots, Portable gadgets, IoT
  - Autonomous Driving, 3D printing
  - Meetups: [grcpp], [Embedded @ Gothenburg]
- Website: https://platis.solutions
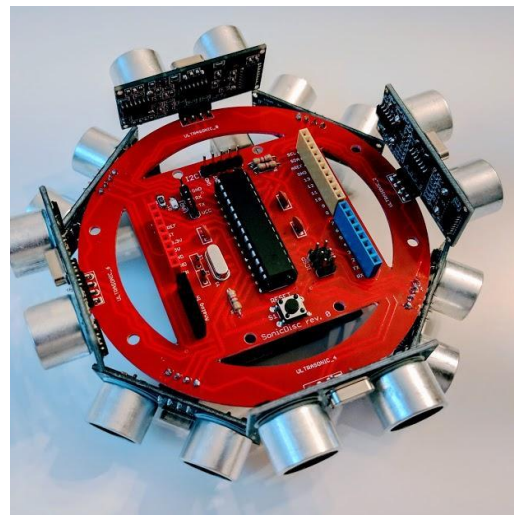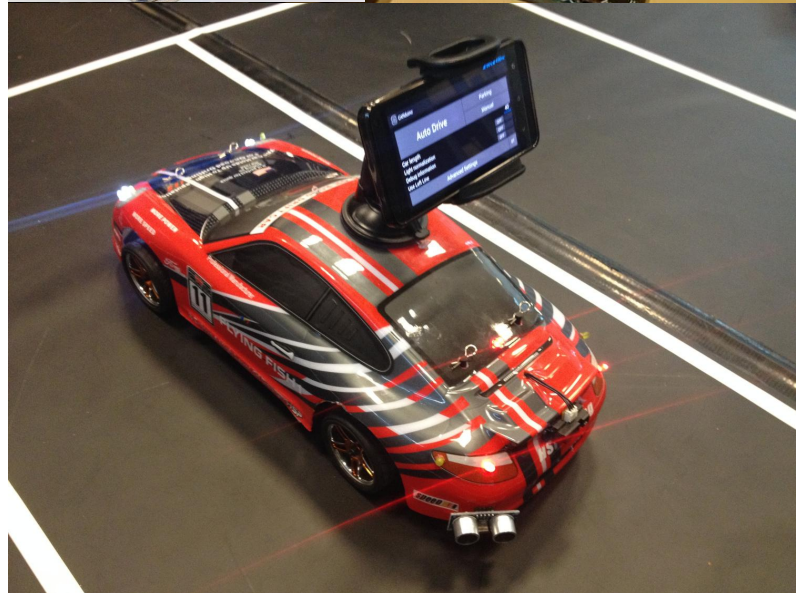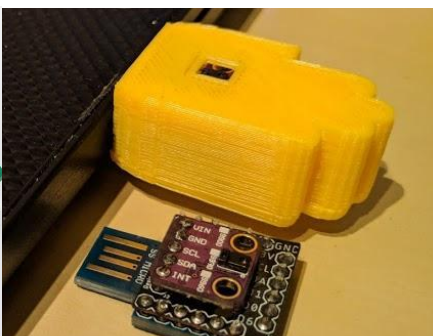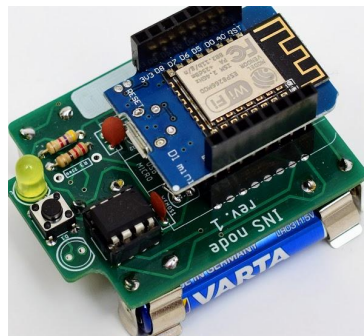
Background

DIT112

DIT112

- Software Engineering & Management BSc
- Compulsory course
- 2nd term
- 7.5 credits
- ex-DIT524 (15 credits)

- ~70 students
- ~12 groups
- Some experience in JAVA
- Have heard of SCRUM
- A bit of experience in git
- A lot of imagination

DIT112 learning outcomes

- Define software in a system context
- Describe system requirements, system and software design, and relations between the requirements and software design
- Organize software development teams and conduct software development projects, using modern software engineering methodologies such as agile development
- Elicit, analyze, and document requirements in the form of a requirements specification

- Design software and document outcome of design work
- Implement software according to a documented software design
- Reflect on integration between software and non-software components
- Evaluate traceability between requirements, design, and implementation artefacts

When software development becomes engineering

It is not about hacking something together that "works", but establishing a development process that is:

- Repeatable
- Defined
- Controlled

Smartcar

A versatile and easy to use vehicle platform
for hobby-grade projects

Smartcar

- Easy-to-use software library
  - Hardware agnostic
  - Support for multiple sensors
- ESP32 microcontroler
  - WiFi
  - Bluetooth
  - FreeRTOS
- L3G4200D gyroscope
- Directional speed encoders
- VL53L0X "micro-LIDAR"
- 5V tolerant I/O pins
- 8 AA batteries
- Open source software & hardware

Switch to Manual Mode

Maneuver the car
Change the speed gear
Increase and decrease the speed
and stop the car

HealthRover
Contactless delivery system

SmartCarEmul
By ItJustWorks™

Magess
DELIVERY SERVICE

YOUR PERSONAL LUGGAGE COMPANION
TRAILGAGE
SMARTCAR
TrailGage

DIT112-V19
DIT112-V20

Challenges

Immature system development process

Sound familiar?

- Scope creep
- Lack of communication
  - Features
  - Defects
  - Vision
- Intermittent quality
  - Frequent regressions

- Lack of domain knowledge
- Untracked work
  - Important for grading
- Unintegrated features

Improving maturity

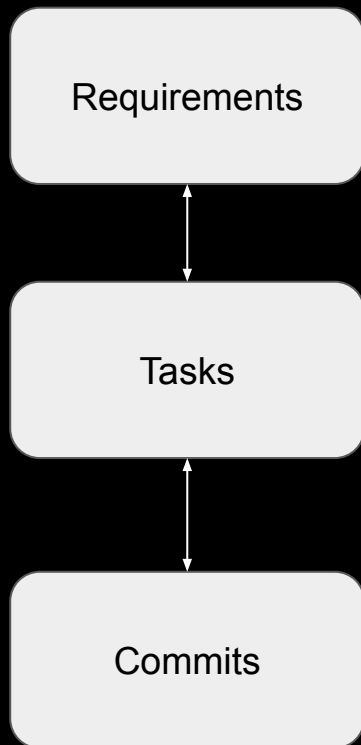Inspired by FOSS development

Working agile

Agile in DIT112

- Customer
- Product owner
- Small & valuable increments
- Weekly sprints
  - Demos
  - Planning
- User stories
  - Persona
  - Acceptance criteria
- PO accepts *only* what is integrated (i.e. on *master*)

Requirements traceability

2-way traceability

Requirements

Tasks

Commits

⇆ Requirements (or Epics)
  ⇆ Tasks (or User stories)
    ⇆ Commits

○ Multiple user stories per epic
○ One epic per user story
○ Link commits to user stories

⇆ Milestones
  ⇆ Issues
    ⇆ Commits

✓ Multiple issues per milestone
✓ One milestone per issue
✓ Link commits and pull requests to issues

Labels used for grouping sprint backlog items

Automated testing

Testing

- Verify requirements
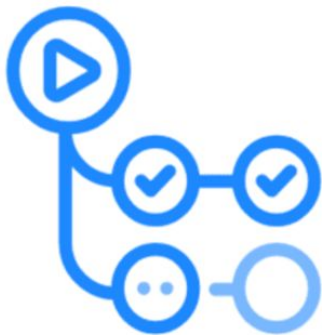- Avoid regressions
- Discover defects before production

Continuous Integration

Automated, defined & continuous

- Build
- Test
- Release
- Deploy

✓ Merge to master allowed only when CI passes
✓ Personal branches ignored
  ○ We don't care about your side-project

GitHub Actions

Documentation

Sustainability & on-boarding

- README.md
  - What/Why/How
  - Demo video
- Wiki
  - User manual
  - Requirements specification
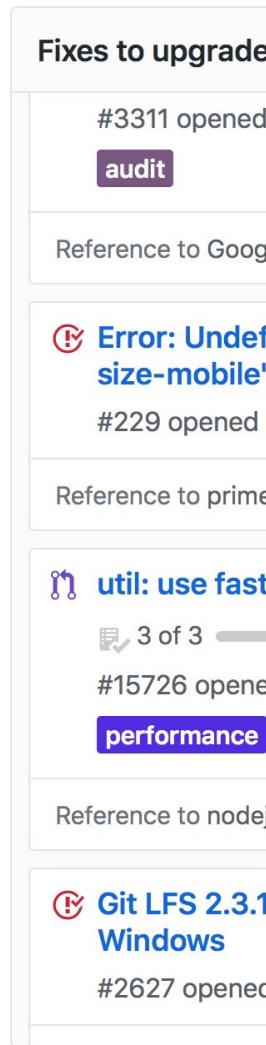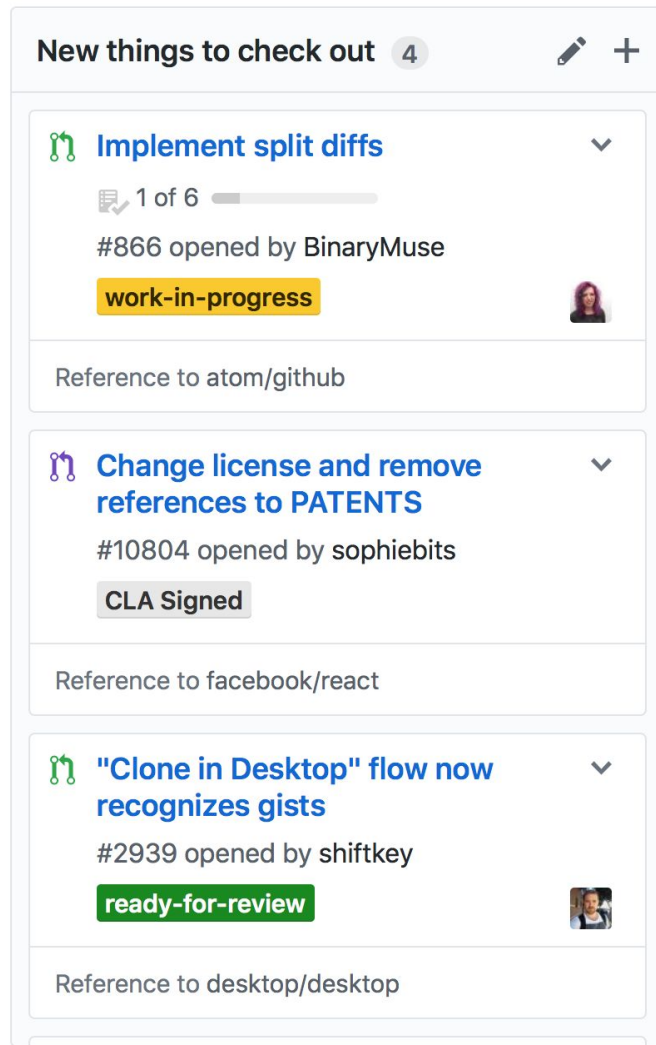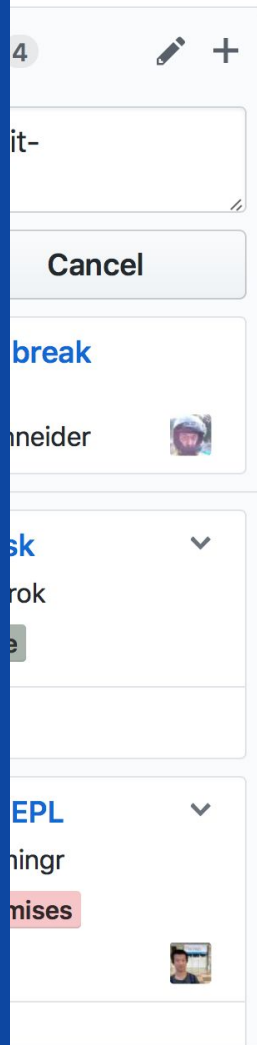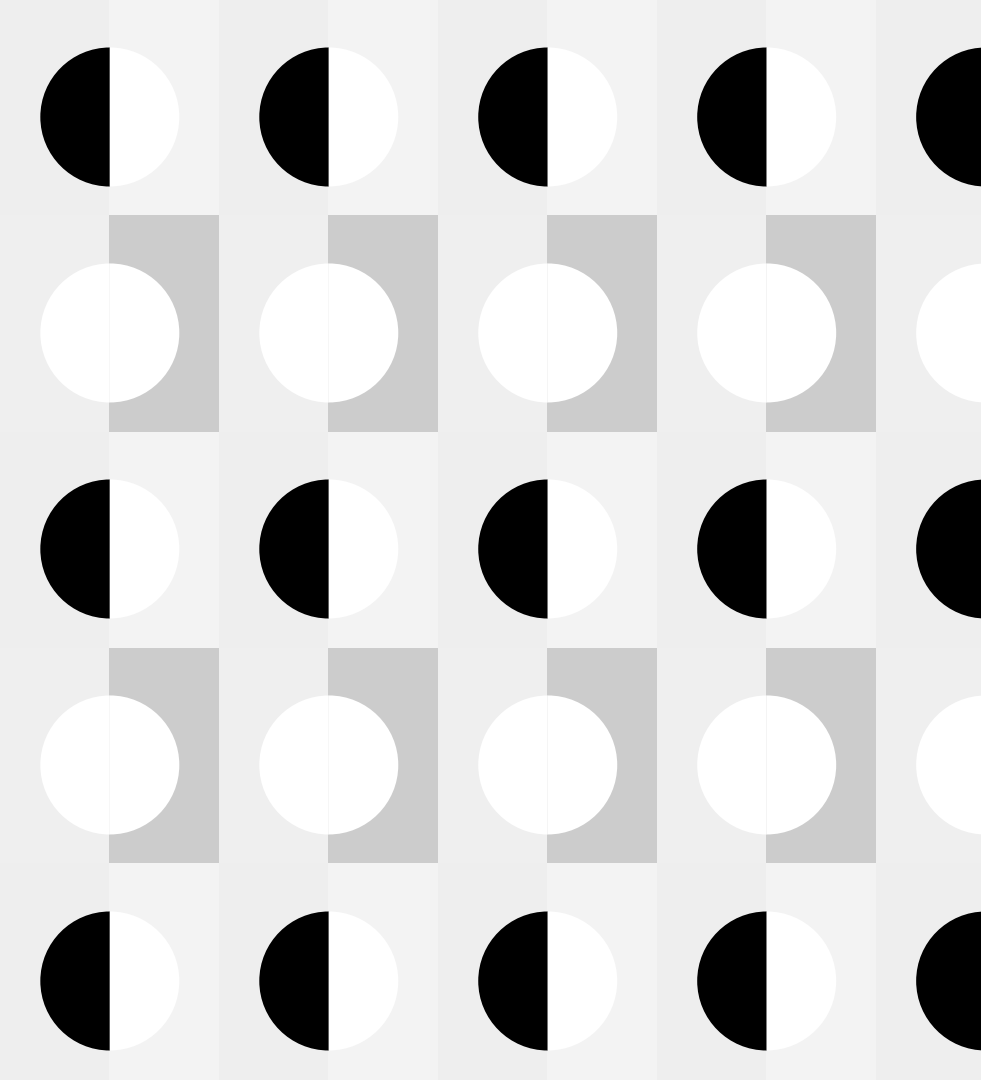- GitHub pages
  - API documentation

Work tracking

Communication & accountability

- Multiple developers assigned on issue
  - Pair programming
  - Developers not *penalized* for collaborating
- GitHub project
  - Issues broken down to tasks
  - Track upcoming, ongoing, finished work
  - Automatically move issues

Cancel

break

neider

sk

rok

EPL

ningr

mises

New things to check out 4

Implement split diffs
📋 1 of 6
#866 opened by BinaryMuse
`work-in-progress`

Reference to atom/github

Change license and remove references to PATENTS
#10804 opened by sophiebits
`CLA Signed`

Reference to facebook/react

"Clone in Desktop" flow now recognizes gists
#2939 opened by shiftkey
`ready-for-review`

Reference to desktop/desktop

Fixes to upgrade
#3311 opened
`audit`

Reference to Goog

Error: Undef size-mobile'
#229 opened

Reference to prime

util: use fast
📋 3 of 3
#15726 opene
`performance`

Reference to nodej

Git LFS 2.3.1 Windows
#2627 opened

Code reviews

Push to master?
No.

✓ Acceptance criteria
✓ Definition of Done
✓ Code review
✓ CI checks

Open development

Peeking is not cheating

- Public sprint demos
  - Short, less than 5 minutes
  - Slides discouraged (only 1 allowed)
  - Live demo if possible
- Public development
  - Solutions to common problems
  - Respect licenses
- Public discussions
  - Canvas LMS
    - Forum
    - Chat
  - Slack

Takeaways

What's your excuse?