Besser Programmieren

T. Himmer (himmerto@hs-weingarten.de)
M. Wydler (wydlermi@hs-weingarten.de)

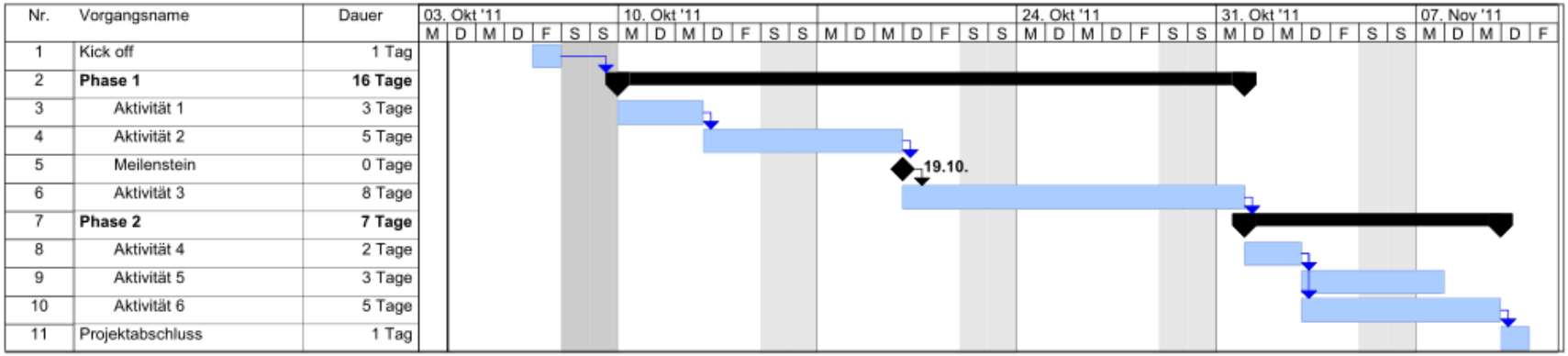
Übersicht

- Organisation
- Versionsverwaltung
- Continuous Integration
- Buildsysteme
- Bibliotheken

Organisation

Organisation

Projektablaufplan (Gantt-Diagramm)



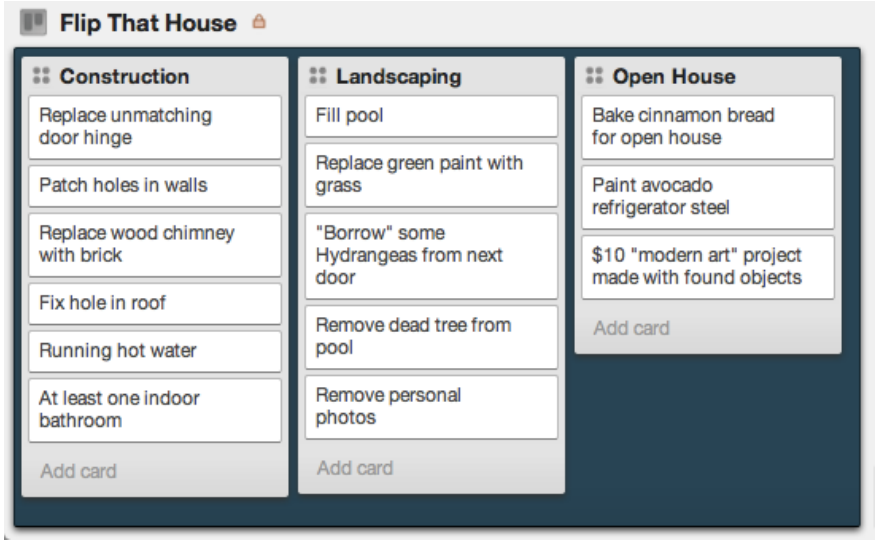
http://de.wikipedia.org/wiki/Gantt-Diagramm

- einzelne Aufgabenpakete
- Aufwand wird geschätzt
- von Ende rückwärts planen
- ist Projekt überhaupt machbar?

Organisation

Aufgabenverteilung (Trello)

- Aufgabenpakete planen
- Entwickler nimmt sich Arbeitspaket
- verschiedene Phasen, *Todo, In Progress, Done*
- jederzeit Überblick über restliche Arbeit/bereits geleistete Arbeit

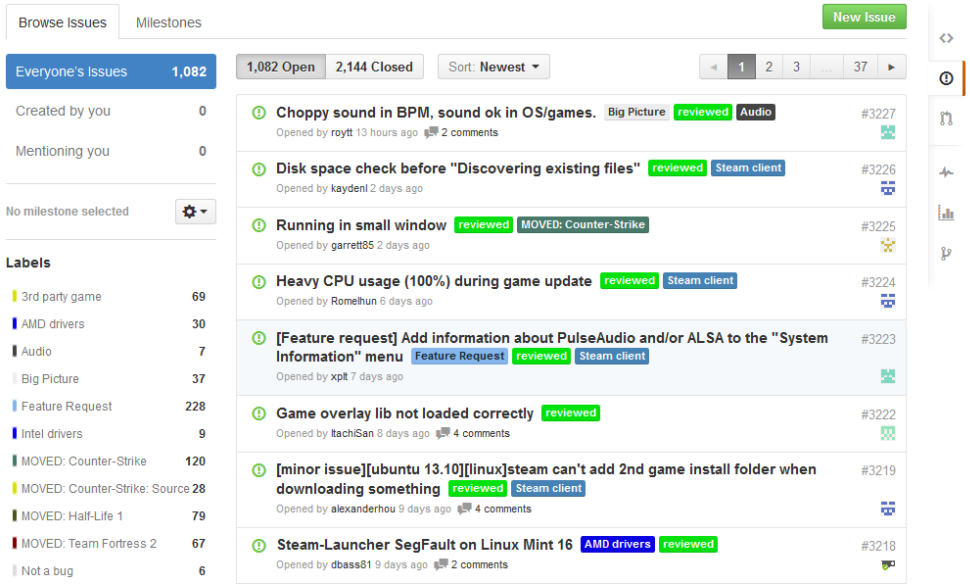


http://static.imgriff.com/1364997960/trello.png

Organisation

Bug-Tracker (GitHub)

- Ticket-System
- Labels (z.B. Bug, Feature, High Prio)
- Personen zuordnen
- Milestones



https://github.com/ValveSoftware/steam-for-linux/issues

Versionsverwaltung

Bazaar

Versionsverwaltung

“Manuelle” Versio[illegible]valtung

- USB-Stick
- Ordner mit _01, _02
- Dateiinhalte selber zu [illegible]nfügen

Versionsverwaltung

- Protokollierung der Änderungen
- Wiederherstellung einzelner Dateien
- Archivierung einzelner Stände
- Koordinierung des Zugriffs mehrerer Entwickler
- verschiedene Entwicklungszweige

Git

Basic Workflow

- Repository erstellen
  `$ git init`
- Datei hinzufügen
  `$ git add [...]`
- Commit der Änderungen
  `$ git commit`
- Status anzeigen
  `$ git status`

Weitere Befehle

- Verlauf anzeigen
  `$ git log`
- Unterschiede anzeigen
  `$ git diff [...]`
- Repository clonen
  `$ git clone [...]`
- Remote bearbeiten
  `$ git remote [...]`
- Repository Pull/Push
  `$ git push [r] [b]`
  `$ git pull [r] [b]`

Git Workflow

1. Schritt

   Repository auf github.com anlegen

2. Schritt

   Repository auschecken

   ```
   git clone [url]
   ```

3. Schritt

   Entwicklungszweig anlegen

   ```
   git branch [branch]
   git checkout [branch]
   ```

Git Workflow

4. Schritt

   Änderungen im Dev-Zweig vornehmen

5. Schritt

   Status des Repository anzeigen

   ```
   git status
   ```

6. Schritt

   Neue Dateien hinzufügen

   ```
   git add [file]
   ```

Git Workflow

7. Schritt

   Dateien committen

   ```
   git commit [-a]
   ```

8. Schritt

   Status des Repository anzeigen

   ```
   git status
   ```

9. Schritt

   Repository pushen

   ```
   git push origin [branch]
   ```

Git Workflow

10. Schritt

    Submodule hinzufügen

    ```
    git submodule add [url] [pfad]
    ```

11. Schritt

    Submodule updaten

    ```
    git submodule update
    ```

12. Schritt

    Aktionen auf allen Submodule ausführen

    ```
    git submodule foreach ‘git ...’
    ```

Continuous Integration

Continuous Integration

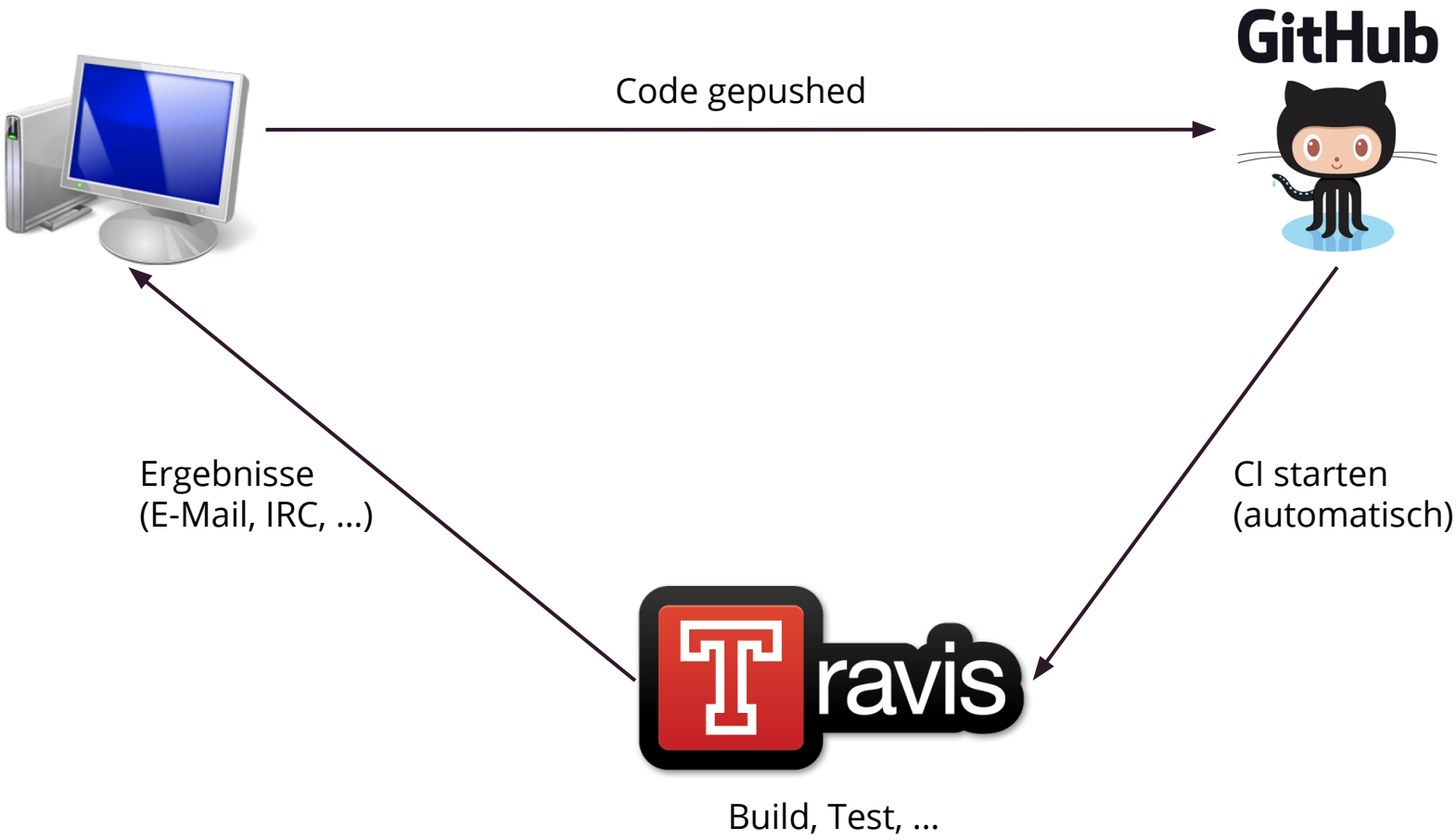
Grundsätze

- gemeinsames Repository
- automatischer Build
- automatische Tests
- häufige Integration
- (Test in Produktionsumgebung)
- (einfacher Zugriff)
- (Ergebnisse teilen)

http://martinfowler.com/articles/continuousIntegration.html

Continuous Integration

Travis-CI



Continuous Integration

Beispiel (travis.yml)

```
language: c

git:
  depth: 1

before_install:
- sudo apt-get update -qq
- sudo apt-get install gcc-avr avr-libc

script:
- (cd wclib && cmake . && make)
- (cd wcfw && make)
```

Build-Systeme

Warum?

- Ermöglicht Continuous Integration
- Plattformunabhängigkeit
- Automatisiert weitere Arbeitsschritte
  - Dokumentation
  - Coding Convention Checks
  - Generieren von Quellcode
  - Testing
  - ...
- Unabhängigkeit von verwendeten IDEs (Meta-Build-System)
- Vereinfacht das Bauen des Projekts

- SCons (www.scons.org)
  - Python
  - Zur Zeit kein Python 3 Support

SConstruct:

```
env = Environment()
env.Program( 'myExecutable', ['main.c'] )
```

SCONS
Build your software, better.

- WAF (code.google.com/p/waf)
  - Python
  - Generiert auch Projectfiles für diverse IDEs

wscript:

```
def options(opt):
        opt.load('compiler_c')

def configure(cnf):
        cnf.load('compiler_c')

def build(bld):
        bld(features='c cprogram', source='main.c', target='app')
```

- Premake (industriousone.com/premake)
  - Lua
  - Generiert auch Projectfiles für diverse IDEs

premake4.lua:

```
solution "MySolution"
	configurations { "Debug", "Release" }

	project "MyProject"
		kind "ConsoleApp"
		language "C++"
		includedirs { "include" }
		files { "src/**.h", "src/**.cpp" }

		configuration "Debug"
			flags { "Symbols" }
			defines { "DEBUG" }

		configuration "Release"
			flags { "Optimize" }
```

- CMake (www.cmake.org)
  - Eigener Syntax
  - Sehr mächtig
  - Generiert auch Projectfiles für diverse IDEs

CMakeLists.txt:

```
cmake_minimum_required( VERSION 2.6 )
project( myProject )
include_directories( include )
add_executable( myExecutable main.cpp )
```

Bibliotheken

Window Management

- FreeGLUT (freeglut.sourceforge.net)
  - Weiterentwickelte Open-Source-Alternative zum veralteten GLUT
- GLFW (www.glfw.org)
  - Display / Window Management
  - OpenGL contexts
  - Keyboard, Mouse, Joystick
  - Clipboard, Time
- SDL (www.libsdl.org)

GL Extensions

- GLEW (glew.sourceforge.net)

- GL3W (github.com/shakesoda/gl3w)
  - OpenGL 3/4 Core Profile only

Image Loader

- FreeImage (freeimage.sourceforge.net)

- ResIL (sourceforge.net/projects/resil)
  - Weiterentwicklung von DevIL (früher OpenIL) (openil.sourceforge.net/)
  - OpenGL-style Syntax

- SDL_image (www.libsdl.org/projects/SDL_image)

Model Loader

- COLLADA (collada.org)

- lib3ds (code.google.com/p/lib3ds)
- Cal3D (home.gna.org/cal3d)

Audio

- OpenAL (kcat.strangesoft.net/openal.html)
  - Eigentlich OpenAL Soft - ein Fork des ursprünglichen OpenAL
  - Bibliotheken für Codecs werden benötigt (z.B.: Ogg/Vorbis xiph.org/vorbis/doc/vorbisfile)

- IrrKlang (www.ambiera.com/irrklang)
  - C++ / C#

- SDL_mixer (www.libsdl.org/projects/SDL_mixer)
  - Für 3D Audio eher ungeeignet

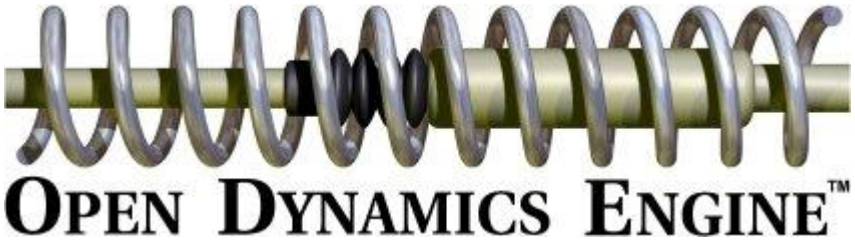
Multimedia

- SDL (www.libsdl.org)
  - Display / Window Management
  - 2D-Grafik, OpenGL Context
  - Simples Audio
  - Keyboard, Mouse, Joystick
  - File I/O, Threads, Timer, Clipboard

- Qt (qt-project.org)

Physics

- ODE (www.ode.org)



- Bullet (bulletphysics.org)

- Newton Game Dynamics (newtondynamics.com)

...

- FreeType (www.freetype.org)
- SDL_ttf (www.libsdl.org/projects/SDL_ttf)
- GLM (glm.g-truc.net)
  - C++ math library for graphics programming

Links

http://github.com

https://travis-ci.org/

https://trello.com/

http://www.git-tower.com/blog/git-cheat-sheet-detail-de/