AdNovum

AdNovum Educatio

The 10 Golden Engineering Rules

Christof Dornbierer, CTO

February 2012

swiss made software

ICSE 2012
June 2-9, 2012
Zurich • Switzerland

Golden Rules – Topics

1. Project Structure / Build Contract
2. QAE / Nightly Build
3. Dealing with Source Code
4. Software Design
5. Security Awareness
6. Usability
7. Packaging / Going Productive
8. Team Culture
9. Technology Management
10. Documentation / Know-how

Project Structure / Build Contract

Rule 1: Always stick to our defined project directory structure and build targets! Automated build on command line has to work (master)!

- **Build targets are a contract for Nightly Build integration**
- **Project has to build automatically using the command line only (IDE build is NOT the master)**
- **Uniform directory structure to navigate and locate artifacts**
- **Separate build artifacts from (versioned) source code**
- **Structure projects into sub components**
- **Provide automated development environment setup (goal: check out, execute additional commands, start working)**
- **Respect customer specifics as well**

→Check out the templates provided with MOAP

QAE / Nightly Build

Rule 2: Test your code and integrate projects into our Nightly Build infrastructure!

- **Think about testability of your code during design (before coding)**
- **Write unit tests to test functionality**
- **Write integration tests to verify whole system/workflows**
- **Perform/request code reviews**
- **Every project has to be integrated in our Nightly Build environment**
- **The Nightly Build has to succeed (if you break it, you fix it!)**

Dealing with Source Code

Rule 3: All code belongs to all engineers, has to be checked in and builds!

- **No check-in of code that does not build locally (on command line)**
- **Update ChangeLog on every commit**
- **No code ownership**
- **No project code outside our central source code repositories**
- **Adhere to our coding guidelines**
- **Think about code generation for redundant infrastructure code**
- **Do not check in generated code and separate from regular code**

Software Design

Rule 4: Don't code without having designed the target system!

- Do macro design (architecture) and micro design (class structure, …) and comment on your decisions
- Keep code and design clean, do refactoring
- Think about re-use and testability
- Think about exception and error handling
- Think about using third-party components (do not reinvent the wheel)
- Simplicity is golden
- Avoid redundancies (also keep generation in mind)
- Stick to well-known patterns

Security Awareness

Rule 5: Always think about the security!

- Always perform input validation/sanitation
- Encode output properly
- Do not leak information (e.g. in case of exceptions)
- Use well-established cryptography (libraries/methods)
- Obscurity is no security
- Do not log critical data (e.g. passwords)
- Implement in-depth security (multiple barriers/checks)
- Do not send critical information by mail
- Do not use unencrypted portable storage

Usability

Rule 6: Think about the users!

- What information is really needed to perform the current action
- Provide proper navigation and navigational hints
- Design workflows as simply as possible
- Think about accessibility
- Think about the presentation of your work result (“It works” doesn't automatically mean “It looks nice”)
- Involve usability@adnovum.ch

Packaging/Going Productive

Rule 7: Employ proper packaging and automated reproducible installations!

- Collaborate with RT to employ proper OS compliant packaging (e.g. RPM for Linux)
- Think about required operation service level and contribute with appropriate operation mechanisms
- Provide a simple command line op interface with the pkg
- Automate installation (no/little interaction)
- Provide configuration in or as a separate package (may include config for prod and non-prod environments)
- Make configuration deterministic (don't rely on default values)
- Think about productive tracing (e.g. ERROR means that immediate action from sysadmin is required)
- Package database setup as well, including DB patching

Team Culture

Rule 8: Ask and take time to answer!

- Ask engineering/support groups or other stakeholders (remember the safaris)
- Take your time to answer questions and support others ("doors are always open")
- Spread your know-how (techznünis, engineering groups, by answering questions)
- Avoid being single point of expertise
- Give constructive feedback (HOW to make it better and WHY)
- Don't take criticism personally, but as a chance to improve
- Projects can only be successful when you're acting as a team

Technology Management

Rule 9: Use only approved technologies!

- Check tactical toolbox and appweb for suitable libraries
- If there's a need for an additional library/tool/technology, ask nerdheads→ technology investigation
- First check open source, think about commercial components and THEN think about coding it by yourself
- Use approved/standardized tooling
- Libraries have to be stored in our central artifact repositories (NEXUS/*pository) and are NOT checked in with the project sources
- Be aware of open source licenses (e.g. viral nature)
- USE the libraries instead of copying/pasting source code

Documentation / Know-how

Rule 10: Document what you do and share your know-how!

- Document what others might use as well (e.g. setup routines, reference environments) → TWIKI
- Documents heading to customers have to be checked by DOC before leaving AdNovum
- Document your code by adding comments
- Share your know-how (techznüni, individually)
- Feed back reusable application patterns to appeng
- Mind our guidelines about document storage (e.g. location, structure, naming)
- Mind also our general naming guidelines (packaging, file names, directories, …)

Summary

- ***Rule 1:*** *Always stick to our defined project directory structure and build targets! Automated build on command line has to work (master)!*
- ***Rule 2:*** *Test your code and integrate projects into our Nightly Build infrastructure!*
- ***Rule 3:*** *All code belongs to all engineers, has to be checked in and builds!*
- ***Rule 4:*** *Don't code without having designed the target system!*
- ***Rule 5:*** *Always think about the security!*
- ***Rule 6:*** *Think about the users!*
- ***Rule 7:*** *Employ proper packaging and automated reproducible installations!*
- ***Rule 8:*** *Ask and take time to answer!*
- ***Rule 9:*** *Use only approved technologies!*
- ***Rule 10:*** *Document what you do and share your know-how!*