WP16

Decisions

Design of Software Architectures

Dr. Vadim Zaytsev aka @grammarware, 12 October 2022

UNIVERSITY OF TWENTE.

Role of the software architecture

ensure to have
sufficient decisions

about the right things

to allow stakeholders
to validate, use, apply them

How?

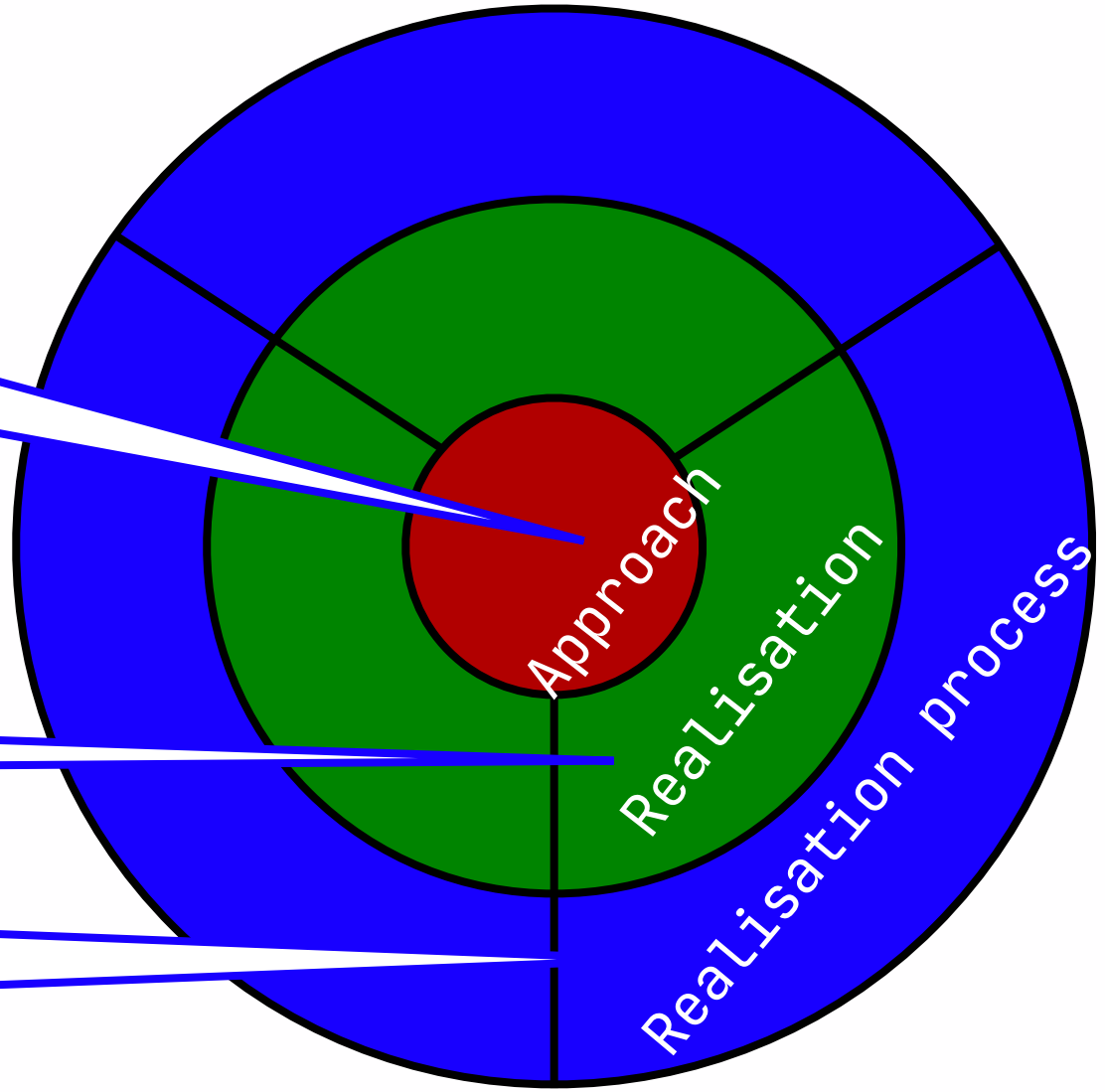
- Standardise: use a fixed architecture framework and fill it
  - standards like ISO 42010, ISO 25010, 4+1, BAPO, CAFCR, …
  - guided by a document template
  - software departments often have one
- Do not think ahead, but let your software creation be…
  - driven by external priorities, set by stakeholders (representatives)
  - iterate over fixed time slots to deal with evolving insights
  - based on experience of developers
- Situational: stakeholder- and concern-driven:
  - select
  - prioritize
  - address
  - iterate…

Realisation domain

By which activities & what behaviour?
- main tasks
- phases
- milestones

Who does what when?
- planning, roadmap,
- responsibilities,
- work packages

Which organisation/people?
- team structure,
- knowledge/skills,
- education/training



by Robert Deckers, based on ISO 42010 & DYA

Ensure the architecture in the development

- Actively:
  - architecture as a requirement
  - do and show it, do it together, let them do it (and monitor)
  - organisationally: by sending architects into development and vice versa
  - code generation / automation
  - standard components
- …defined by the process
- Guarding:
  - testing (functionally and code)
  - review the code (detailed design)
- …defined by the process

Guidelines for giving direction

The system must be reliable.	*Use classes in your Java code.*
The system must be secure against unauthorised users	*The system has to be modular.*
Make sure the system is reusable.	*Each student should have 2+ campus activities/week.*

Guidelines for giving direction

negation is nonsense	generic truths
inconsistent in itself	effortlessly fulfilled
not understandable	impossible to enforce

Decisions!

- The development activity is basically a group “actors” taking decisions that:
  - happen in certain context:
    - time: during a project, in maintenance, in a sprint…
    - space: specific persons belonging to the development activity
  - matter:
    - decision topic is relevant
    - remain (in specifications and heads)
  - finally result in a *formal* machine-readable specification
    - source code
    - configuration settings
    - rules, models

Who decides what and when

- Who are the deciding actors?
- What do they decide about?
- When are those decisions on the timeline?

- What happens when you swap an early & a late decision?