
SA Stories
• in-house build frameworks
• building an enterprise system is a complex activity and training the staff on every detail of each adopted framework is expensive
• companies try to simplify and standardise the usage of adopted frameworks by building their own super-framework
• this allows to:
• guarantee a certain level of homogeneity (e.g. transaction handling, data transfer, UI style, etc..)
• simplify development tasks
• enforce a certain architecture (for both structure and behaviour)
• in-house build frameworks are defined based on past experience coming from different projects
• this is just one way to reduce the gap between planned and actual architecture. Usually integrated with other architectural validation techniques
• generic code quality tools as main entry point for architectural and design rules conformance
• rules are clearly specified in the documentation but checking if the system is consistent with them is expensive
• architects use the hints provided by generic code quality tools as starting points for deeper code review
• reasons could be that adequate tools for guiding the reviewer towards potential architectural anomalies are still missing (supports research goal) and the definition of project specific rules related to design and architecture is still complicated and not well supported.
• example: Sonar + Findbugs/Checkstyle
• in-house modeling and verification tool
• one company has developed a quite sophisticated modeling tool which encourages architects to encode parts of their specification according to a more formal language (based on a graphical notation; boxes and arrows)
• the tool is able to check if certain properties specified in the model have been correctly implemented in the code
• examples:
• physical deployment of sw components (specified graphically and checked by querying the configuration management db)
• interfaces (specified graphically and checked through static analysis of source code)
• dependencies between components (specified graphically but still not checked -> components identification is based on naming conventions and not on the structure of packages. This prevents them from adopting publicly available tools like Jdepend)
• problems:
• tracing of entities
• developing new analysis is very expensive
• sometimes publicly available analysis tools do not match internal conventions