Software Architecture in an Agile World

IOANNIS KOSTARAS

AGILE CRETE 2016

What about architecture?

- How does it comply with the agile world?
- Is it in conflict?
- Agile Manifesto

Agile in the Customer world

- Customers are interested in 3 things:
  - How much it will cost
  - What the product will look like
  - When it will be done
- However, agile stresses:
  - Agility - responsiveness to change and uncertainty
  - Not accurate high-level planning

Incompatible with agile methodologies

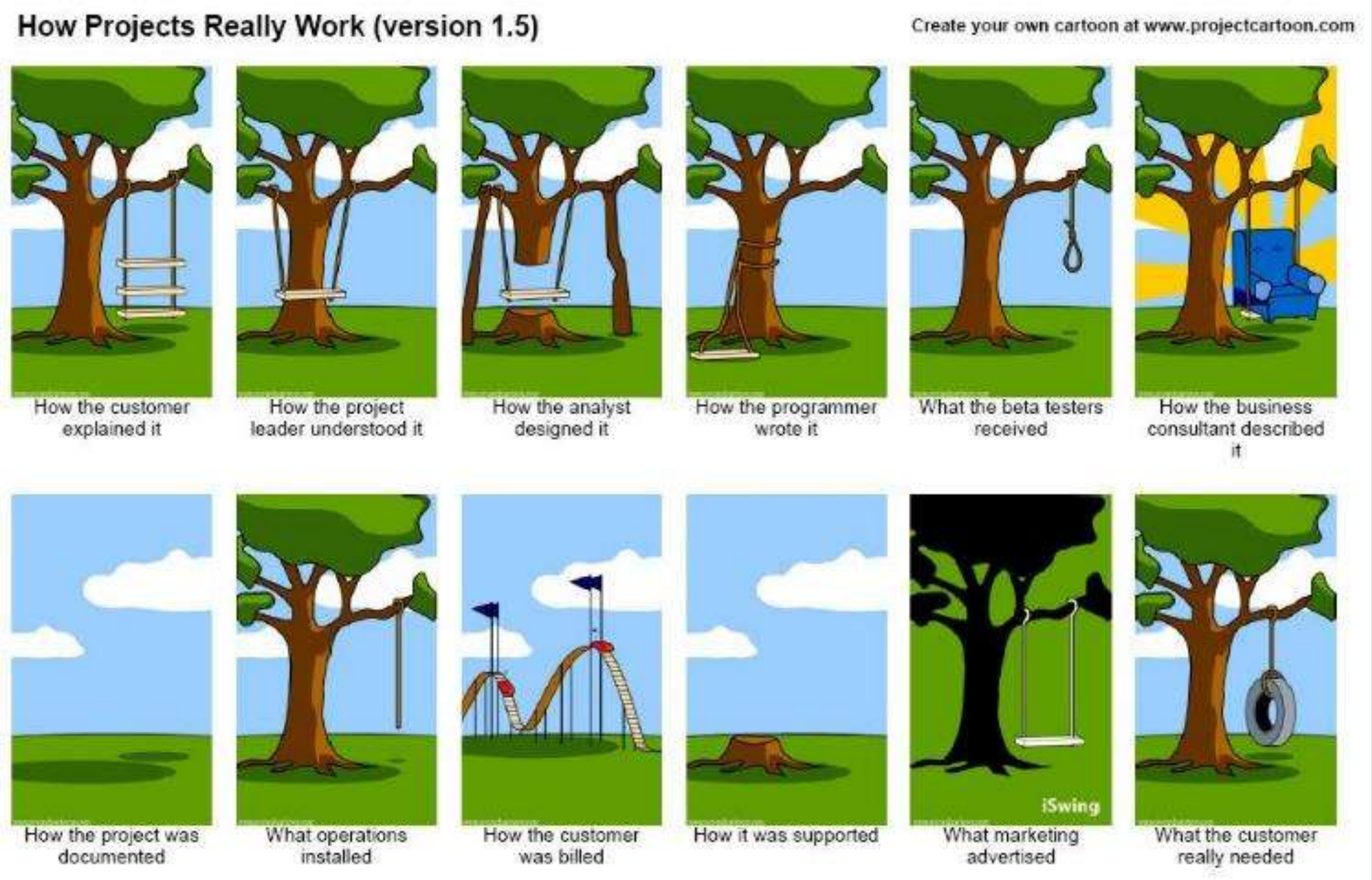
What about architecture (cont.)?

- Team structure:
  - agile software development teams strive towards reducing the amount of overhead associated with communication via document hand-offs.
  - In agile, self-organised, cross-discipline teams the traditional role of a dedicated software architect who traditionally produces a mass amount of architectural/design documents is not very in favour in the team.

What about architecture (cont.)?

- Process and output:
  - The traditional approach to software architecture, that of a big design up front to settle on an understanding of everything that needs to be delivered before putting a blueprint (and usually a plan) in place, is contrary to one of the key goals of agile approaches to deliver customer value, frequently and in small chunks.
  - Agile teams find TDD as an excuse to avoid big up front design.
  - TDD is *not* a substitute for software design.
  - The process of thinking about software architecture is really about putting some boundaries in place, inside which you can build your software using whatever xDD and agile practices you like.

What about architecture (cont.)?


How Projects Really Work (version 1.5)
Create your own cartoon at www.projectcartoon.com
How the customer explained it
How the project leader understood it
How the analyst designed it
How the programmer wrote it
What the beta testers received
How the business consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
iSwing
What marketing advertised
What the customer really needed

What about architecture (cont.)?

- Architecture is about the stuff that’s hard or costly to change. The trick here is to differentiate what’s important from what’s not. Defining a high-level structure to put a vision in place is important.
- How much up-front design is enough? Agile methodologies don’t say “don’t do architecture”.
- The thing is that you need to do “just enough” up front design to give you structure and vision in order to communicate what is your goal and how you can achieve it.
- An agile architecture can be implemented using e.g. microservices and be component-based so that components can be built independently and according to the agile principles.

How to introduce software architecture to your team?

- Educate people
- Talk about architecture in retrospective
- Add software architecture to your definition of done
- Allocate the software architecture role to a team member
- Remember that spending too much time on coding you lose the big picture; spending too much time on architecture you do no coding at all.

“Working software over comprehensive documentation”

- These words do not mean “don’t write documentation at all”.
- The underlying principle here is that real working software is much more valuable to end-users than a stack of comprehensive documentation but many teams use this line in the agile manifesto as an excuse to not write any documentation at all.
- Unfortunately the code doesn’t tell the whole story and not having a source of supplementary information about a complex software system can slow a team down as they struggle to navigate the code base.
- Modern tools like AsciidocFX (for Asciidoc) and PlantUML can help incorporate documentation inside the code instead of delivering big Word documents.

“Working software over comprehensive documentation” (cont.)

- Create drawings (e.g. UML diagrams) or sketches to visualize the architecture but don’t do any big design upfront
- Make sure that code and software architecture are aligned (e.g. by adopting an architecturally-driven coding style).
- Think of the missing documentation as a guidebook, which should give people enough information to get started and help them accelerate the exploration process of the software.

References

- Brown, S. (2016), *Software Architecture for Developers*, LeanPub.

Questions