

A STORY FROM

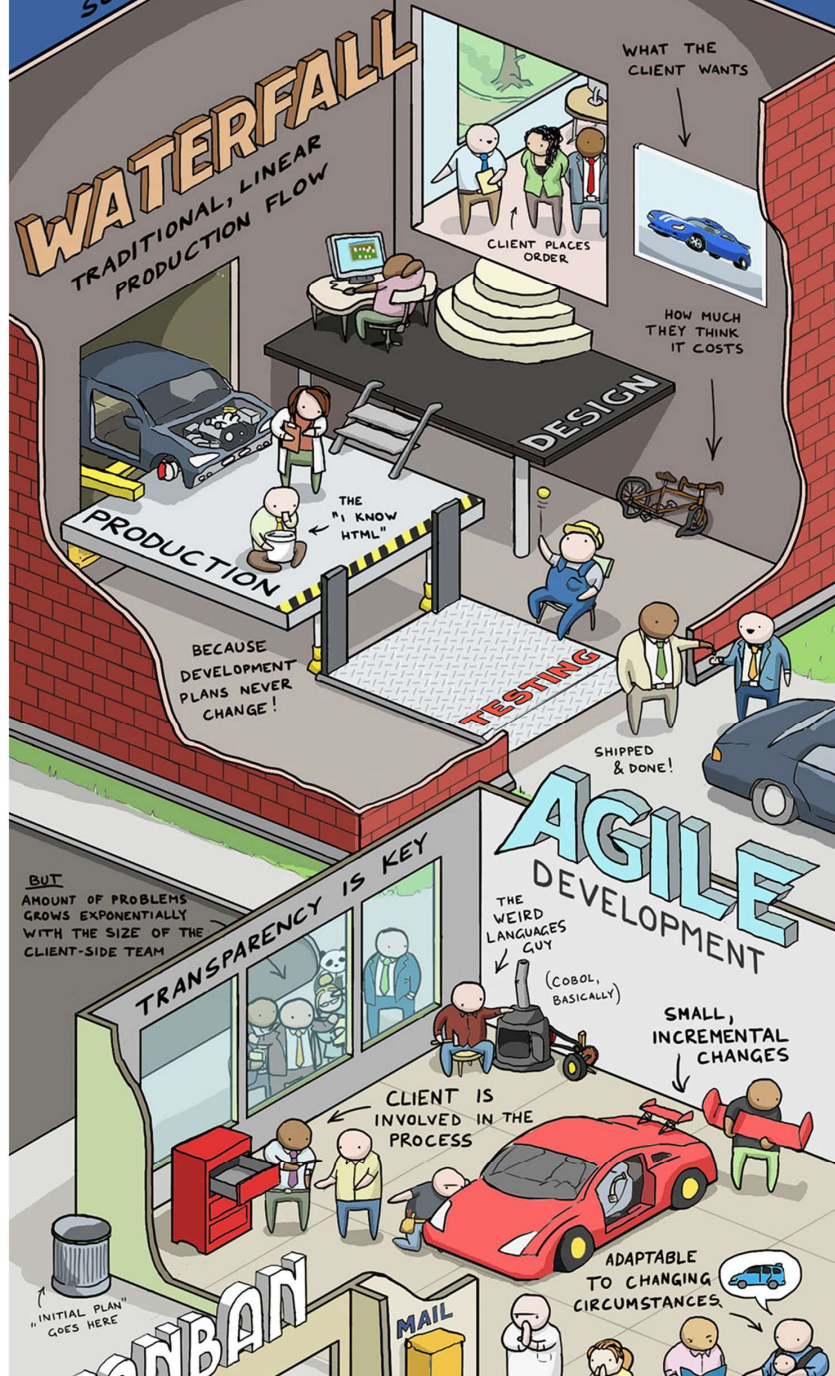


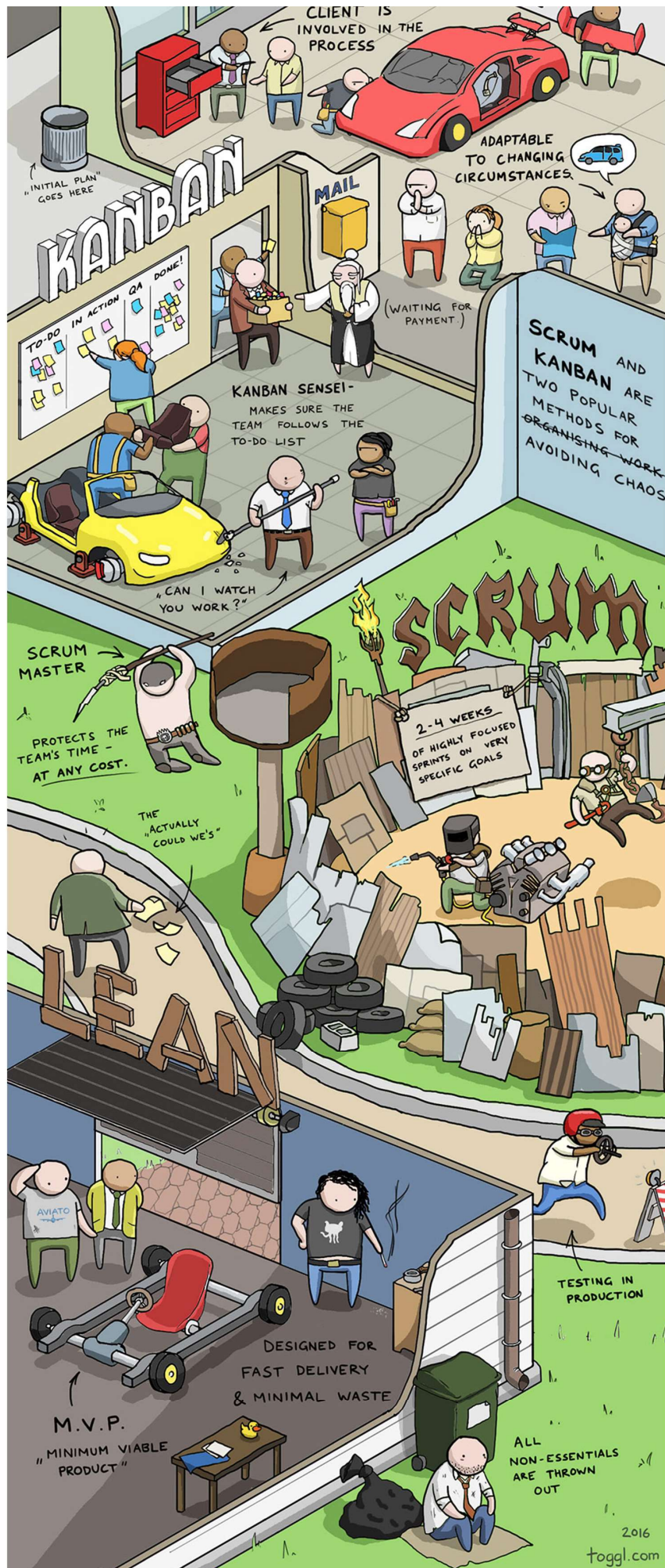
BY MART VIRKUS

SOFTWARE DEVELOPMENT

EXPLAINED WITH CARS

THE NUTSHELL VIEW INTO SOFTWARE AGENCY MODELS





AGILE

THE BIG PICTURE

SCRUM IS GREAT FOR:

People

- Team decision making
- Team empowerment
- Lots of communication
- Promotes simplicity, courage, feedback and respect



Management

- Priorities set by business value
- High task & progress visibility
- Formal processes
- Fosters continuous improvement



SCRUM DOES **NOT** DEFINE **HOW TO CODE**. FOR THAT YOU NEED **EXTREME PROGRAMMING**.

TEST-DRIVEN DEVELOPMENT

Query Capture

As a call centre agent I want to capture the client name, email and cell number so that I can follow up the client's query

TESTS

- Email required
- Must have @
- No spaces
- ...

write test that fails
↓
refactor
↺ *write code to pass test*

WHY? TO LOWER DEFECT RATES.

PAIR PROGRAMMING

- Two heads are better than one
- Simple solutions from the start (good ✓)
- High intensity with peers delivers productivity benefits

**WHY?
EVEN LOWER
DEFECT RATES.**



DOMAIN DRIVEN DESIGN

Build to last • Design for ease of understanding
Speak business's language • Easy to add on or modify
Reduce rework and misunderstandings

HOW?



Understand the business domain



Model each part



Code & test application layers

NEXT STEPS

1. Find a good architect to coach your team
2. Attend a course



KRS
KHANYISA REAL SYSTEMS
Software Since '87

The **ONLY** Agile Development Course In South Africa
ADVANCED AGILE DEVELOPER COURSE

CAPE TOWN 12-13 MARCH 2013

021 681 2900 www.krs.co.za training@krs.co.za



SCRUM

VS

KANBAN

VS

SCRUMBAN

PLANNING → REGULAR

occurs at the beginning of sprint

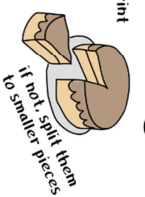


ESTIMATIONS OF TIME

DEFINE start of sprint



items should be small to finish within sprint



ROLES

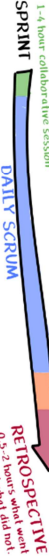
CHANGES TO WORK SCOPE

should wait for next sprint

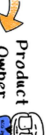


MEETINGS

SPRINT PLANNING
1-4 hour collaborative session



OWNERSHIP

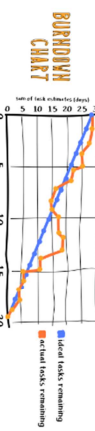
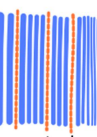


WHEN TO USE

Small items - small value
adding increments is possible
requirements are clear
more cross-dependent teams

BOARDS / ARTIFACTS

PRODUCT BACKLOG
SPRINT BACKLOG
TO DO
DEV
TEST
DONE

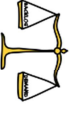


PLANNING

NOT PRECISE planning routine

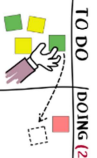
PLAN WHEN THEY FINISH items

demand planning



ESTIMATIONS OF TIME

optional when items are completed



teams simply pull next item from backlog and implement it

ROLES

CHANGES TO WORK SCOPE

added as needed



MEETINGS

NONE REQUIRED

DEPENDS on defined roles and necessities

OWNERSHIP

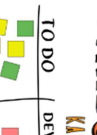
Who me?

WHEN TO USE

Changes are too fast
support/maintenance work (operational level)
bugfix

BOARDS / ARTIFACTS

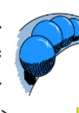
KANBAN BOARD
TO DO
DEV (3)
TEST (4)
DONE



PLANNING

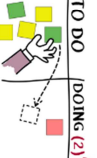
plan when items are completed

iterations can eventually stay



ESTIMATIONS OF TIME

optional



teams simply pull next item from backlog and implement it

ROLES

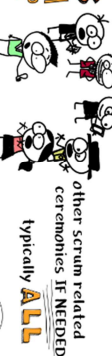
CHANGES TO WORK SCOPE

added as needed



MEETINGS

DAILY SCRUM STANDUP



OWNERSHIP

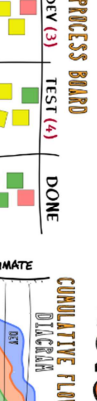
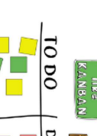
Yes me!

WHEN TO USE

no clear roadmap
or
too fast changes
Need to include support/maintenance (event driven) work in the process

BOARDS / ARTIFACTS

PROCESS BOARD
TO DO
DEV (3)
TEST (4)
DONE



WATERFALL VS. AGILE

Which project management style is right for you?

Sometimes, the secret to a successful project is selecting the right project management style. Establishing an effective framework is crucial for any project to run smoothly. But when you have **WATERFALL** and **AGILE** planning methods to choose from, how do you know which is the best for your project and team?

Here's a list of the pros and cons for each method to help you decide.

WATERFALL



Waterfall requires detailed planning at the beginning of a project.



All the steps are laid out, dependencies mapped, and you move to the next stage only after completing the previous one.

Pros.



Best for projects that deal with physical objects – from a construction project to a hardware installation project.



Best for projects with defined tasks and phases that must be completed in a specific sequence (e.g., build the first floor of a building before the second floor).



Project plans are repeatable for identical or similar projects in the future.

Cons.



Requires substantial scope and schedule planning before work begins.



Scope changes can be slow and require formal change control processes.



Less effective for software, design and other non-physical or services-based projects.

AGILE



This is a fast and flexible approach to project management based on principles of collaboration, adaptability and continuous improvement.



Unlike the orderly stages of a waterfall approach, agile project management is typically set up in quick, iterative project release cycles.

Pros.



Best for projects that deal with services-oriented and non-physical deliverables like code, copywriting and design projects.



Allows for quick course correction based on stakeholder feedback.



Empowers project teams to work creatively and efficiently.



Includes engagement and collaboration from all team members.

Cons.



Not suited for projects with strictly defined requirements and scope.



Uncertainty around scope and schedules can make stakeholders and executives nervous (at first).



Requires vigilant backlog and documentation maintenance, and tech debt management.

The project management method you choose will vary based on the project, your team and goals. Once you select a planning style, make sure you use project management software that lets you and your team set up your projects the way you want. Luckily for you, LiquidPlanner supports waterfall, agile and even ad hoc solutions to fit your needs.

HAPPY PLANNING!