

Save money by avoiding waste in Software Development

Alliance Software

Who am I?

Alex Green - Project Manager and Workflow Fanatic

What are we going to talk about?

Waste - What it is, and why you should really care about it!

Software Development Methodologies - The bedrock of what we do

Kanban - The single most useful tool you've ever met

The Perfect Project - The Holy Grail, and how to find it

Why do we care about waste?

- No system or process is perfect
- If we can get better at seeing waste and assessing it, we can save both time and money



So, what is waste?

Waste (noun) - a software terminology thing

Any effort which occurs on a project which doesn't result in value for the customer

Seven Types of Waste

- Work in Progress
- Overengineering
- Hand offs
- Task Switching
- Delays
- Relearning the process
- Defects



Work in Progress

Perhaps the most prevalent form of waste, but one of the easiest to tackle.

Work which has been partially (or fully completed) but has not been released to customers.

An Example

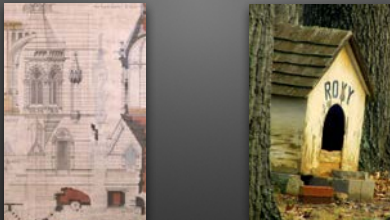


Overengineering

Doing more work than was required to achieve a certain outcome.

The heart of beating this form of waste is active cost-benefit analysis.

An Example



Hand Offs

The waste involved when a task has to be handed over from one worker in a process to another.

The new person has to get up to speed with the story so far.

An Example



Task Switching

The battle of responsiveness versus efficiency.

Swapping from one task to another, then having to swap back to the original.

An Example



Delays

Waiting between the end of one process and another.

Indirect waste, with an impact on elapsed time, rather than effort.

An Example



Relearning the Process

Needing to get back up to speed on a piece of the system which has not been looked at for a while.

An Example



Defects

Issues in the code itself.

Can be caused by:

- Process not being followed
- Spec misunderstood or not followed
- Insufficient or incorrect spec
- Indirect effects of new code
- Human error

An Example



So, what is waste? (reprise)

Any effort which occurs on a project which doesn't result in value for the customer

- Work in Progress
- Overengineering
- Hand offs
- Task Switching
- Delays
- Relearning the process
- Defects

How does waste apply to my project?

Every project has waste - I know mine do. The first step is acceptance.

Examine your system and identify waste.

Then amending the system to account for that waste.

RBWET's

Ready-baked waste evasion techniques



Task Switching

- If the interruptions come from external clients, increase the price of interruptions to ensure this are truly critical
- Have a 'support' team and a 'feature' team, the former shielding the latter
- Work in 'chunks' of ~3 hours, and never break a chunk.

Relearning the Process

- Batch your work into features/ fixes which are alike
- Document where required, and where useful

Defects

- Increase the rigour of the rules in your system
- Introduce new work phases to address the types of defects
- Improve communication between phases and workers



Software Development Methodologies
Dogmatism versus Pragmatism

What is a
software
development
methodology?



Easy, it's...

- A process or series of processes used in software development
- Generally taking the form of defined 'phases', such as 'design', 'testing' etc.
- Designed to describe the 'how' of the lifecycle of a piece of software

Some
examples?
...there can't be that
many, surely...

- Agile Software Development
- Crystal Methods
- Dynamic Systems Development Model (DSDM)
- Extreme Programming (XP)
- Feature Driven Development (FDD)
- Joint Application Development (JAD)
- Lean Development (LD)
- Rapid Application Development (RAD)
- Rational Unified Process (RUP)
- Scrum
- Systems Development Life Cycle (SDLC)
- Waterfall (a.k.a. Traditional)

- Agile
- Lean
- Waterfall



The Lie

Perhaps the biggest single lie when talking about software development methodologies is...

...that there is one, outright best way to do it...

...and that if it doesn't produce a good result result then you didn't do it right.

Waterfall

- Sequential
 - *The phases occur in a defined sequence, one after the other*
- Non-iterative
 - *The phases occur only once in the lifecycle of the product; no repetition*



What's Waterfall good for?

- Very clear on outcomes
- Sign off from large numbers of stakeholders
- Forces decisions early
- Allows for estimation/quoting up front

The downsides of waterfall

- Humans can be pretty bad at thinking of the full picture
- The larger the project, the more scope there is for change
- No ability to change the product to meet a changing market
- Doesn't allow for a change in the process; even if you spot areas for improvement

Agile

- Adaptive
 - *The process and the product are expected to change as the project advances*
- Collaborative
 - *A focus is placed upon communication between team members*



What's Agile good for?

- Allows you to 'pivot' around issues e.g. a competitor entering the market
- Increases your speed to market
- Spreads your revenue more evenly across the project
- Highly visible process and outcomes

The downsides of Agile

- Requires active engagement from all members
- Flexibility can lead to misalignment of ideas and miscommunication
- The continuous nature of development can feel like progress isn't being made

Lean

- Waste-averse
 - *Deliver in the most light-weight manner possible*
- Iterative
 - *The process and product are incrementally improved via cycles of development and learning*



What's Lean good for?

- Addresses risks quickly and cheaply
- High focus on market validation and making a successful product, not the 'right' product
- Empowers the team, improving engagement

The downsides of Lean

- Requires excellence from the team members
- Requires buy-in from stakeholders and commitment to the concept
- Requires flexibility in the team

Why are you trying to define a process?

This is not a trick question!

By defining a process, you're...

- Providing a set of rules
- Creating a shared understanding in your team
- Improve the quality of your product
- Remove or alleviate risks in your project



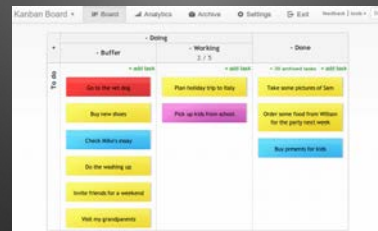
Defining the best process

- As lightweight as it can be
- Satisfies the needs of the project
- Engages your team



Kanban board (todo, doing, done)
Kanban - Learning from the Japanese

What is Kanban?



What is Kanban?

- A scheduling system designed for just-in-time manufacturing
- A board (physical or electronic) with a set of structured columns



What is Kanban?

- Later phases pick up work from earlier phases
- Most simplistic representation would be To Do, Doing, Done



The Heart of Kanban

...as used by Alliance Software



How we use Kanban

- Improves visibility
- Helps us define the rules of the development cycle
- Keeps us asking the question 'why?'
- Denote and promote ownership
- Encourage conversation and communication



Why would you use them?




Why would you use one?

- By having a concrete process, you can start to improve it
- They improve the visibility of all ongoing work
- Improve accountability



Running the Perfect Software Project

The Team (Alliance Software)




Alliance Software

Alliance logo, or team photo

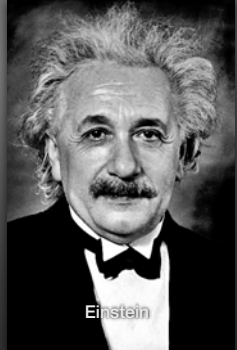
What's needed?

- Friendly and accommodating
- Empowered to make decisions and raise issues
- Flexible and non-dogmatic
- Solution-driven
- High degree of ownership



Same as previous slide

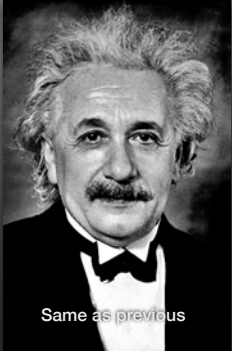
The Team (the Client)



Einstein

What's needed?

- Open to discussion
- Engaged in the process
- Motivated to succeed in their market
- A single point of contact



Same as previous

The Process



apple computer inc.

Blueprints

What's needed?

- A starting point
- Agreement from all parties
- Know where the dials are
- Visibility



apple computer inc.

Same as previous

What we've learned

- What waste is, the common waste types found in software development, and how to recognise and tackle them
- We've explored a number of software development methodologies and their strengths and weaknesses
- What a Kanban board is, and how the Kanban practices can enhance a project