Raising the bar:

Super optimizing your Agile implementation using Kanban and Lean

Jesper Boeg, Trifork Software Pilot
jbo@trifork.com

Guilherme Silveira,
Tech leader at Caelum

November 3, 2010

In general

- Let us know if:
  - You have questions (The most important thing is not covering every single slide)
  - What we are saying does not make any sense at all
- This is the first time we do this presentation so any feedback will be much appreciated

Agenda

- Agile anti patterns
- We can do better!
- Lean principles can help us understand how
- Implementing lean using Kanban
- How does that fit with traditional Agile practices?
- Difficulties
- For mature teams only? (If time permits)

AGILE ANTI-PATTERNS

TRIFORK.

1. Breakdown madness

- Items small enough to fit a 2 week iteration are often too small to deliver real business value
  - Test becomes waste
  - Retrospectives become waste
  - Feedback becomes waste

Breakdown Madness

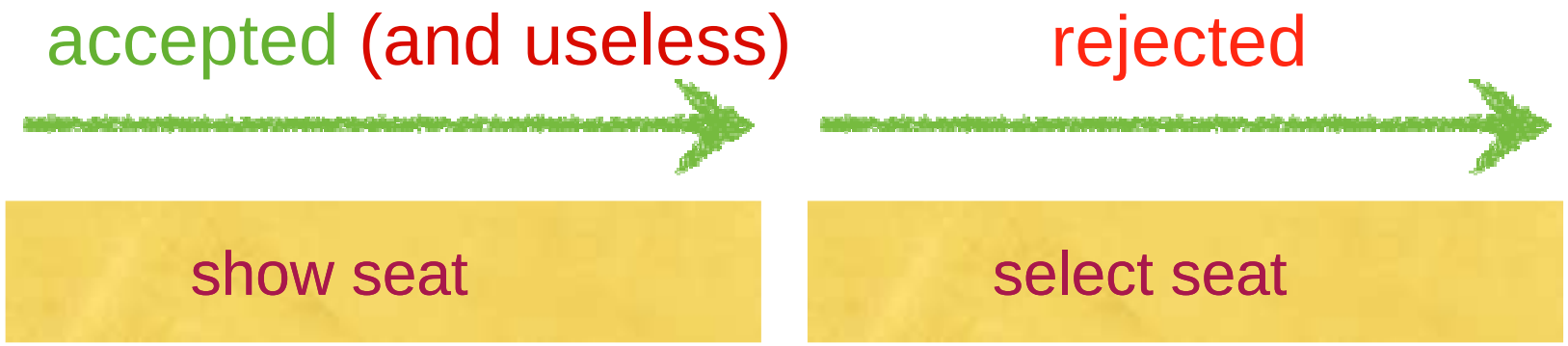
support booking seats

Breakdown Madness

show seat

select seat

Breakdown Madness



good sense of half victory
fake sense of half victory

2. Sprinting to meaningless deadlines

- Fixed iteration goals stress the entire system:
  - Product owners rush to prepare for upcoming cycles
  - Testers race to complete work late in the development time-box
  - Developers prioritize finishing a set of features over refactoring, TDD and pair programming

Sprinting (continued)

- Goal is to get everybody working at 100 capacity 90 percent of the time
- Disregard for the cost of running a unit at 100 percent utilization
  - Flexibility
  - Quality
  - Flow restriction
  - Unsustainable pace

3. Cargo Cult batch sizes

- God told us that all Agile projects will fit 2-4 week development cycles without:
  - Regard to business
  - Deployment cost
  - Feedback quality
  - Competition
  - Minimal marketable features

4. Sub optimization

- Lack of focus on the entire delivery
  - Development is by definition seen as the bottleneck
  - Clear definition of roles restrict flexibility
  - PO, Development, Test and Operation become silos
  - Protecting the sprint without considering the consequences

Restricted Roles

I do it because a wise man once said:

“This is how things were done before.
Don’t ask questions, just do it.”

Restricted Roles

Adapt as required:

roles
technologies
tools
methodology

5. Synchronizing everything

- Prioritization, delivery, inspection, reflection and planning are synchronized to maximize periods of undisturbed work
  - Delaying feedback
  - Reducing flexibility
  - Applying the lowest common denominator

6. Too much work in progress

- Enormous backlogs
- All items on the sprint backlog in progress
- **Almost**: tested, reviewed, released

I am almost

ignorant wise

guilherme

i can code and solve some simple problems

I am almost

poor rich

guilherme

i can eat and attend one conference an year

I am almost

ugly

a top model

guilherme

i can date almost anyone i want

I am almost

idle

done

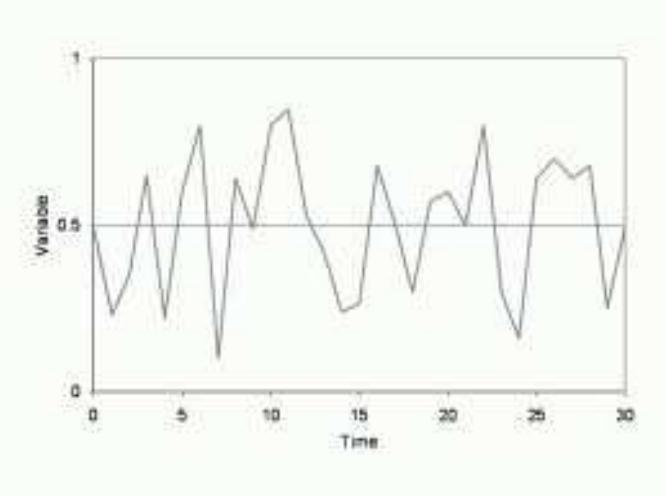
feature

It’s not finished: you can not benefit from it!

1 done > 10 useless

7. Variability reduction

- Reducing variability to increase predictability
  - Leveling story -size, -complexity, -risk
  - Ignoring the nature of product development
  - Ignoring the cost of wanting 100 percent short term predictability

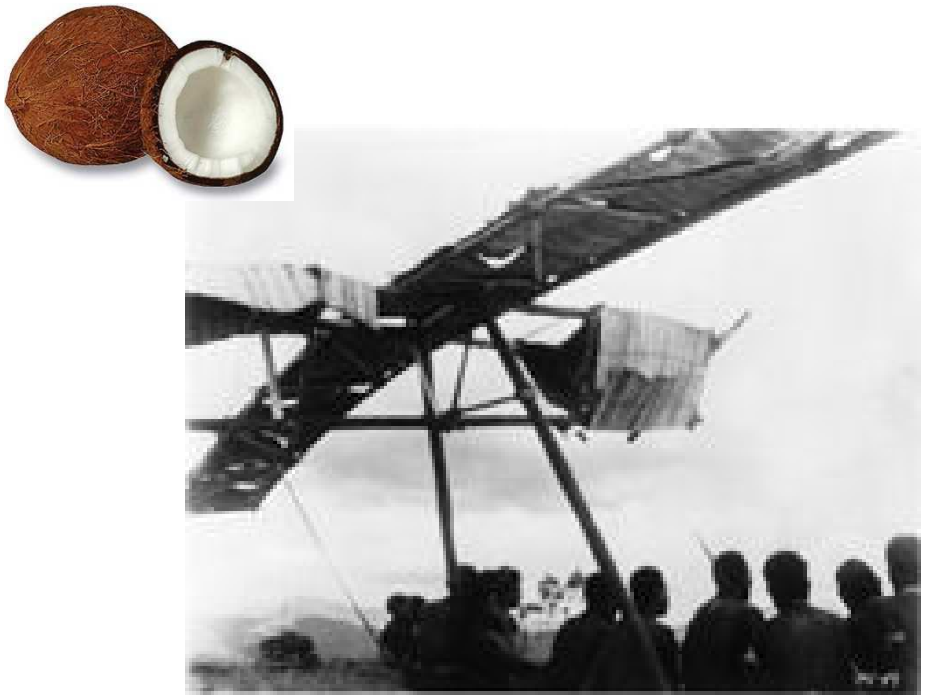


WE CAN DO BETTER!

AND FORTUNATELY A FEW SIMPLE LEAN PRINCIPLES CAN HELP US UNDERSTAND WHY

First we must remove ourselves from faith based Cargo Cult implementations

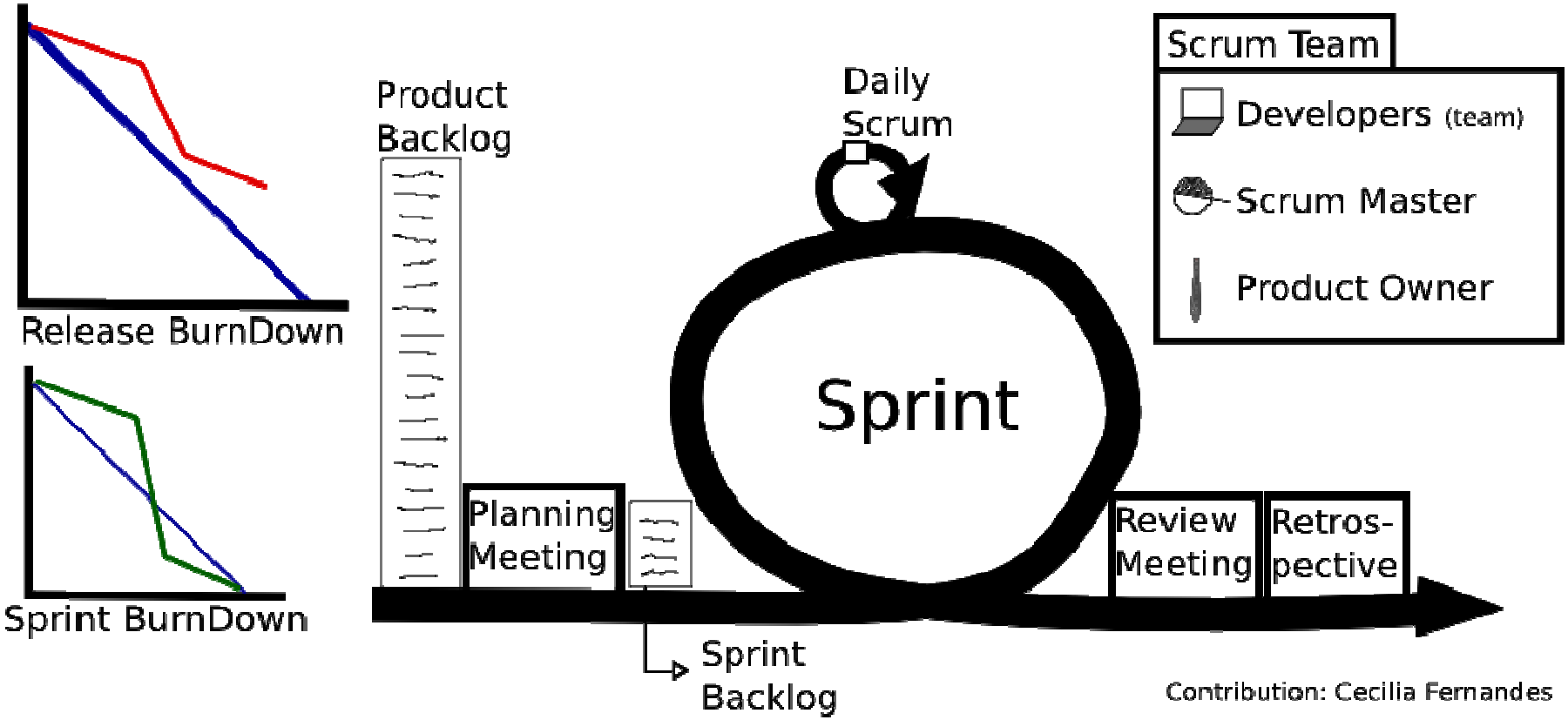
- Once practices become faith based and cargo cult we risk loosing sight of the goal



The goal is not:

- Iteration retrospectives
- Perfect sprint burn downs
- Frozen iterations
- Clear separation between PO and SM
- 2-4 week iterations
- Prioritized backlogs
- Excellent story point estimates

if everything is rigid


Release BurnDown
Sprint BurnDown
Product Backlog
Daily Scrum
Scrum Team
Developers (team)
Scrum Master
Product Owner
Sprint
Planning Meeting
Review Meeting
Retros-pective
Sprint Backlog
Contribution: Cecilia Fernandes

options?

But could be:

- Giving customers the best possible ROI
- Establishing flow across the entire value chain – to reduce WIP, increase flexibility and time to market
- Fast Feedback loops - to facilitate early and continuous improvement
- Quality built in - to increase trust, reduce rework and the cost of bug administration
- Defer decisions – to reduce queues, wasted effort and embrace change
- Valuing people over processes and tools – to make the best use of the resources at hand according to the specific context

8 LEAN PRINCIPLES

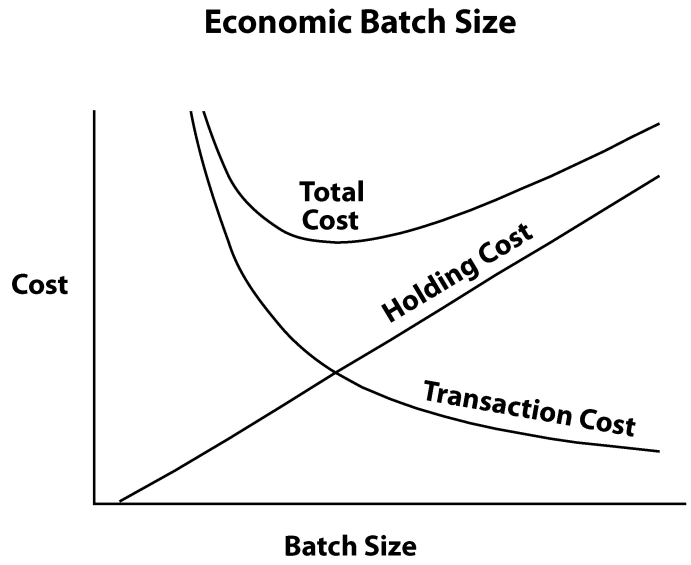
Lean principle 1

- Identify value from the customer's perspective

Breakdown madness occurs when story point throughput becomes more important than customer value

Lean principle 2

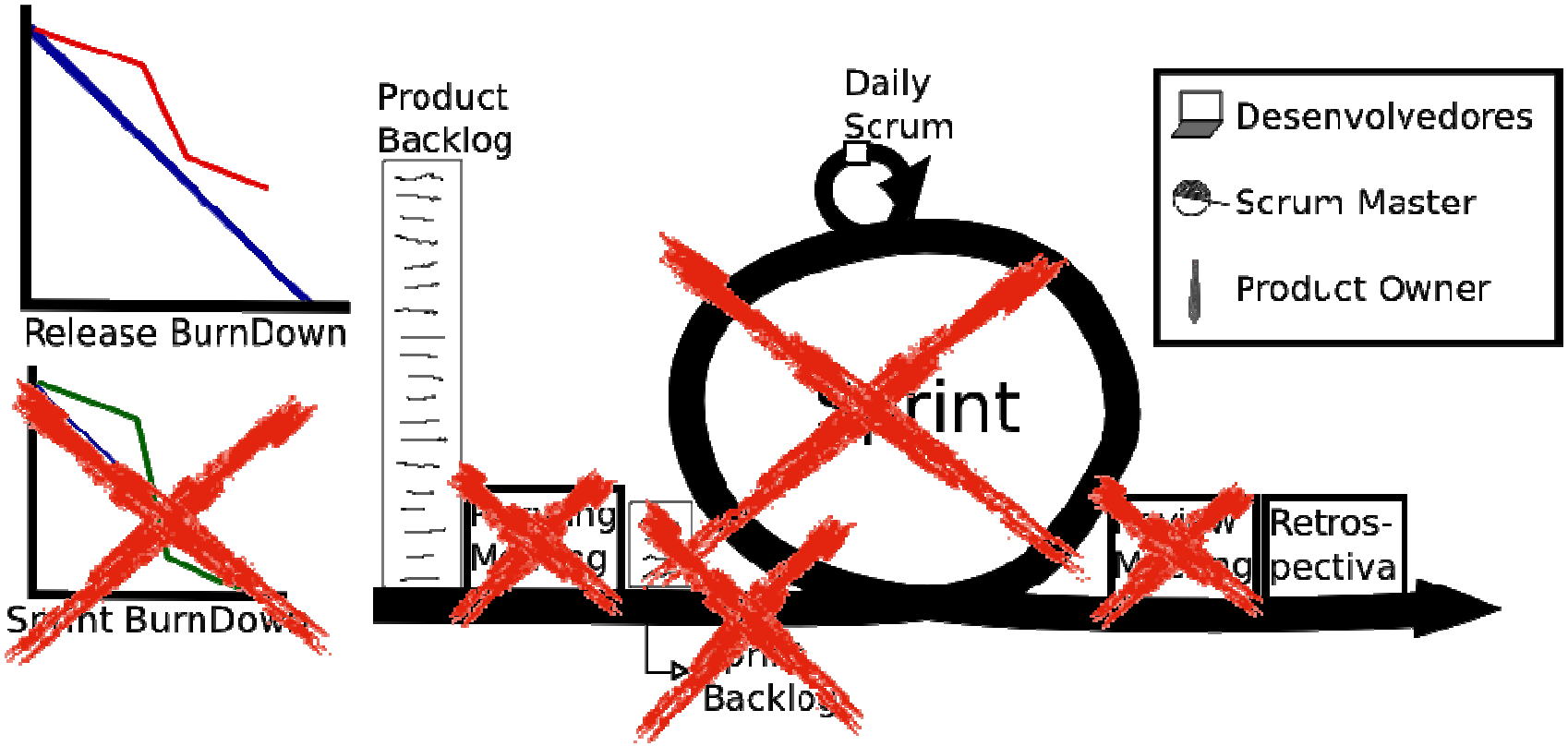
- Batch size is a U-curve optimization, not faith based and context independent.



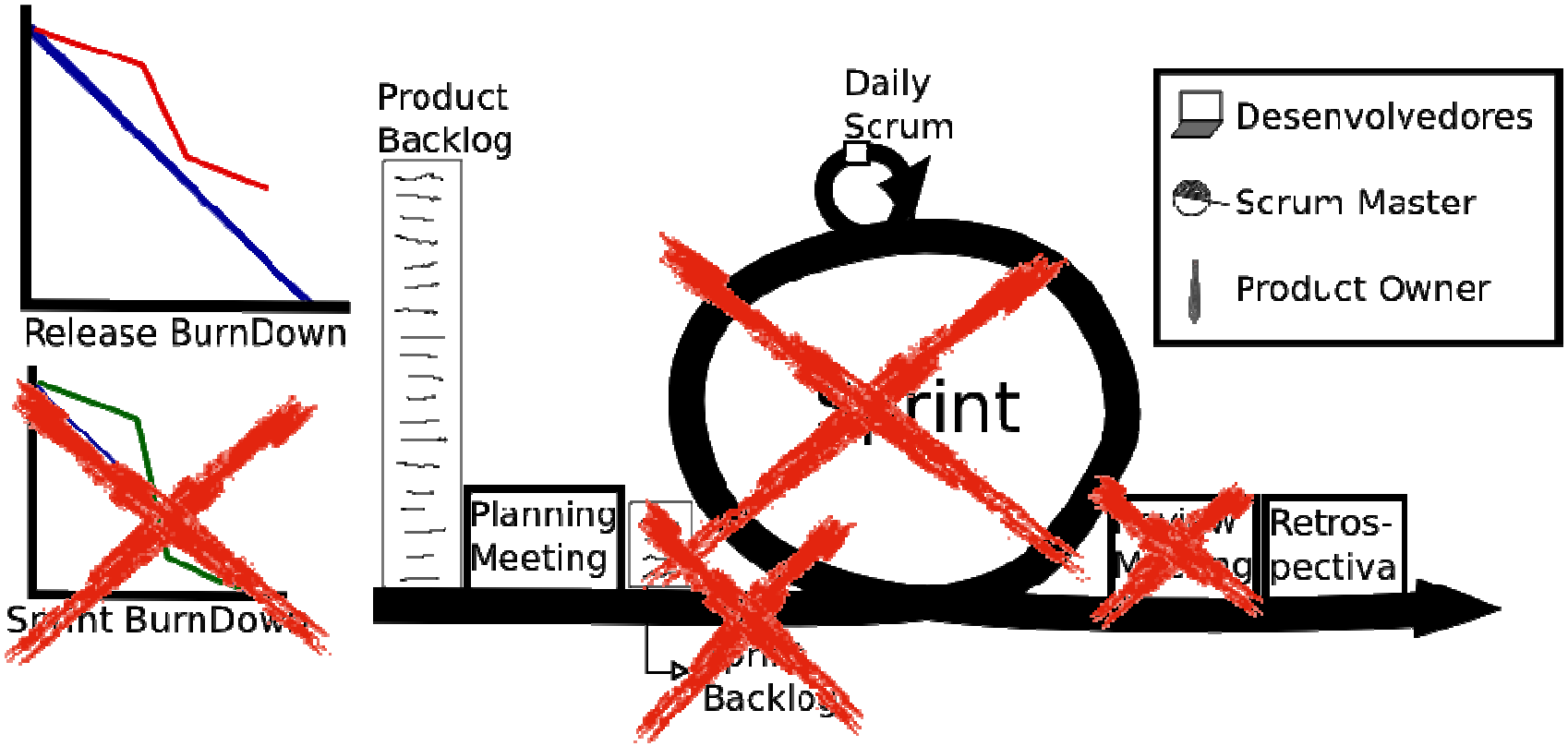
From "The Principles of Product Development Flow," by Donald G. Reinertsen. Celeritas Publishing: 2009. Copyright 2009, Donald G. Reinertsen

But Toyota taught us that transaction costs are not fixed

you may drop it



you may drop it



Lean principle 3

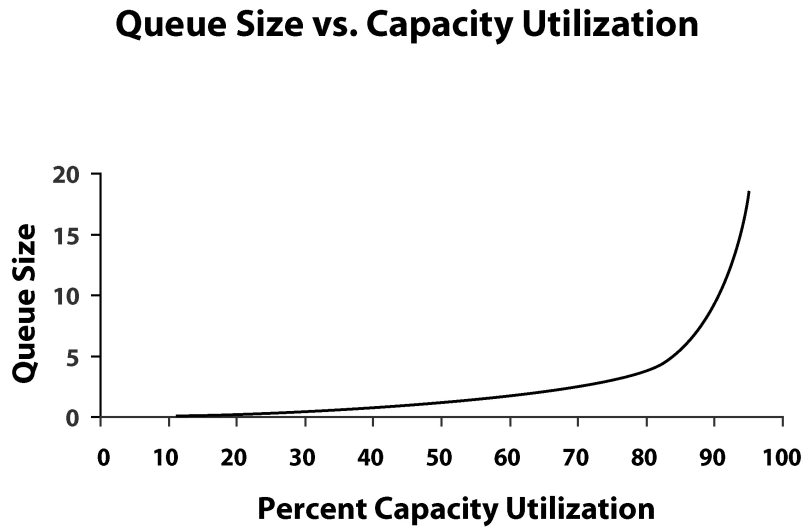
- Optimize queues, batch sizes and flow – not utilization. Don’t stress the system to increase capacity utilization:
  - Stressing the system leads to unsustainable pace, low quality and increased breakdowns in the production flow
  - Focusing on detailed plans in high variability environments like product development reduces the chance of pooling variability and leveling throughput
  - Identify the cost of expediting

Lean principle 4

- Optimize the whole and invest in flexibility
  - Development is not always the bottleneck and sub optimization will often stress the real bottleneck even more
  - Flexibility is often a good investment in high variability environments like Product Development
  - Focus on effective silos is batch optimization
- Though queues off the critical path are not free knowing the critical path is very important

Lean principle 5

- Maximum capacity utilization is a failure mode



From "The Principles of Product Development Flow," by Donald G. Reinertsen.
Celeritas Publishing: 2009. Copyright 2009, Donald G. Reinertsen

Lean principle 6

- Let the cadence vary according to context and transaction costs
  - Kanbans are used in Lean manufacturing because machines work with different cadences and synchronizing everything in one piece flow is not the optimal solution
  - Synchronizing everything creates blocks, bottlenecks and queues.
  - Especially in high variability environments

Lean principle 7

- Reduce WIP to improve flow, feedback and time to market
  - Measuring and controlling queues are far more approaches effective than reducing variability and improving capacity

Lean principle 8

- We cannot create value without variability in product development
  - Therefore we should always seek the optimal balance between expected payoff and probability

Choise	Cost	Payoff	Probability	Value
A	20.000	60.000	30%	12.000
B	30.000	60.000	50%	15.000
C	50.000	60.000	90%	9.000

INTRODUCING KANBAN

Copy machine

Paper inventory

No paper! Order something!

1 day later

Tell me you need paper before you need it!

Order 7

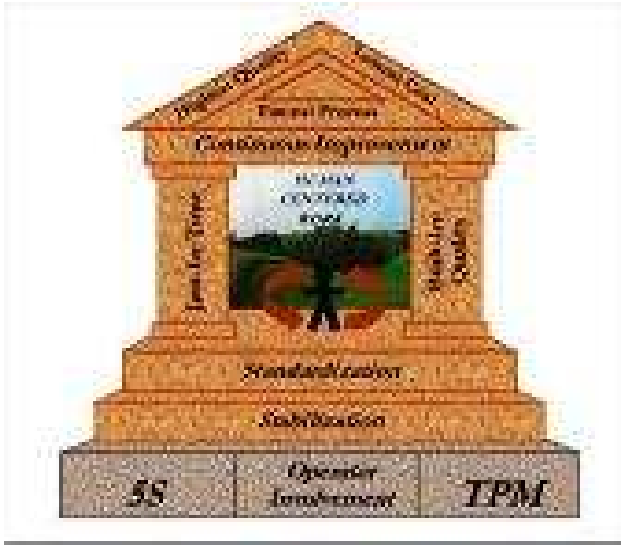
A simple example of a Kanban pull system

- New paper is ordered when the limit prescribed by the Kanban is reached
- When paper arrives the Kanban is returned along with the paper

KANBAN PROVIDES US WITH A SIMPLE SET OF PRINCIPLES TO APPLY LEAN TO SOFTWARE DEVELOPMENT

Limit work in progress

- Focus on flow not utilization
  - Focus and motivation can be achieved without stressing the system with meaningless deadlines
- Deliver often to gain early feedback



Quality built in

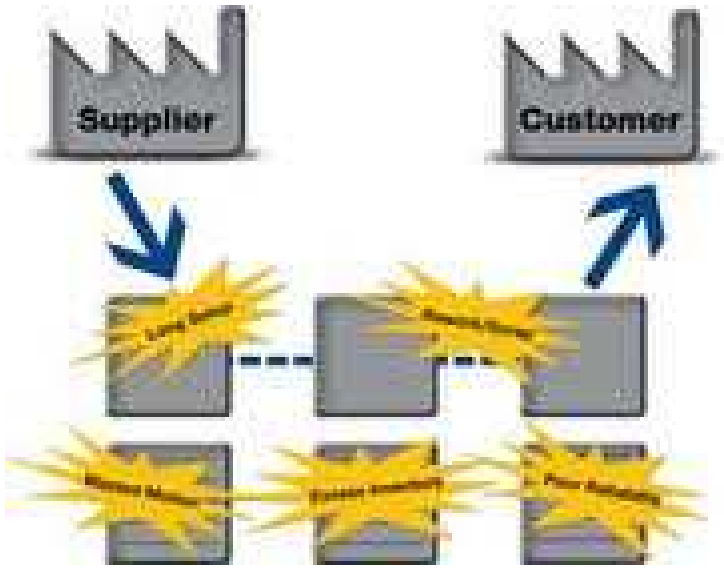
- Stop the line mentality

Continuous improvement

- Part of the culture and a state of mind

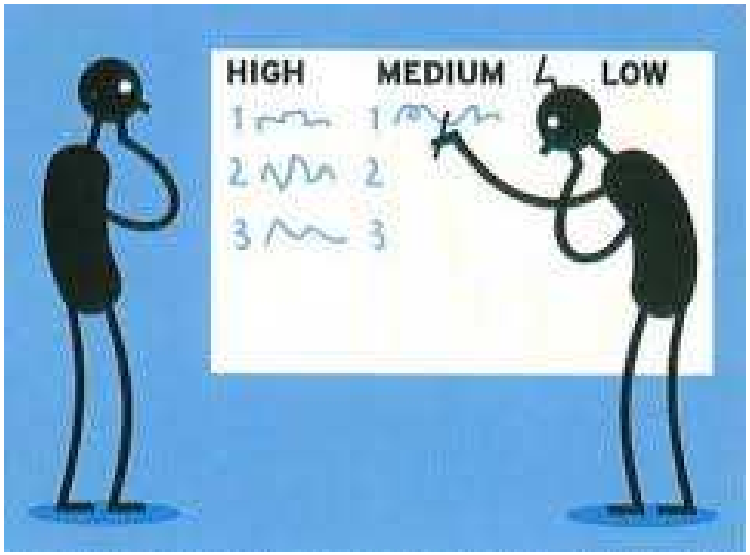
Optimizing the whole

- Balance demand and throughput
  - Sustainable pace – no “cell” should work at more than 80-85 percent capacity
  - Having free time on your hands
  - Optimizing the whole



Prioritize

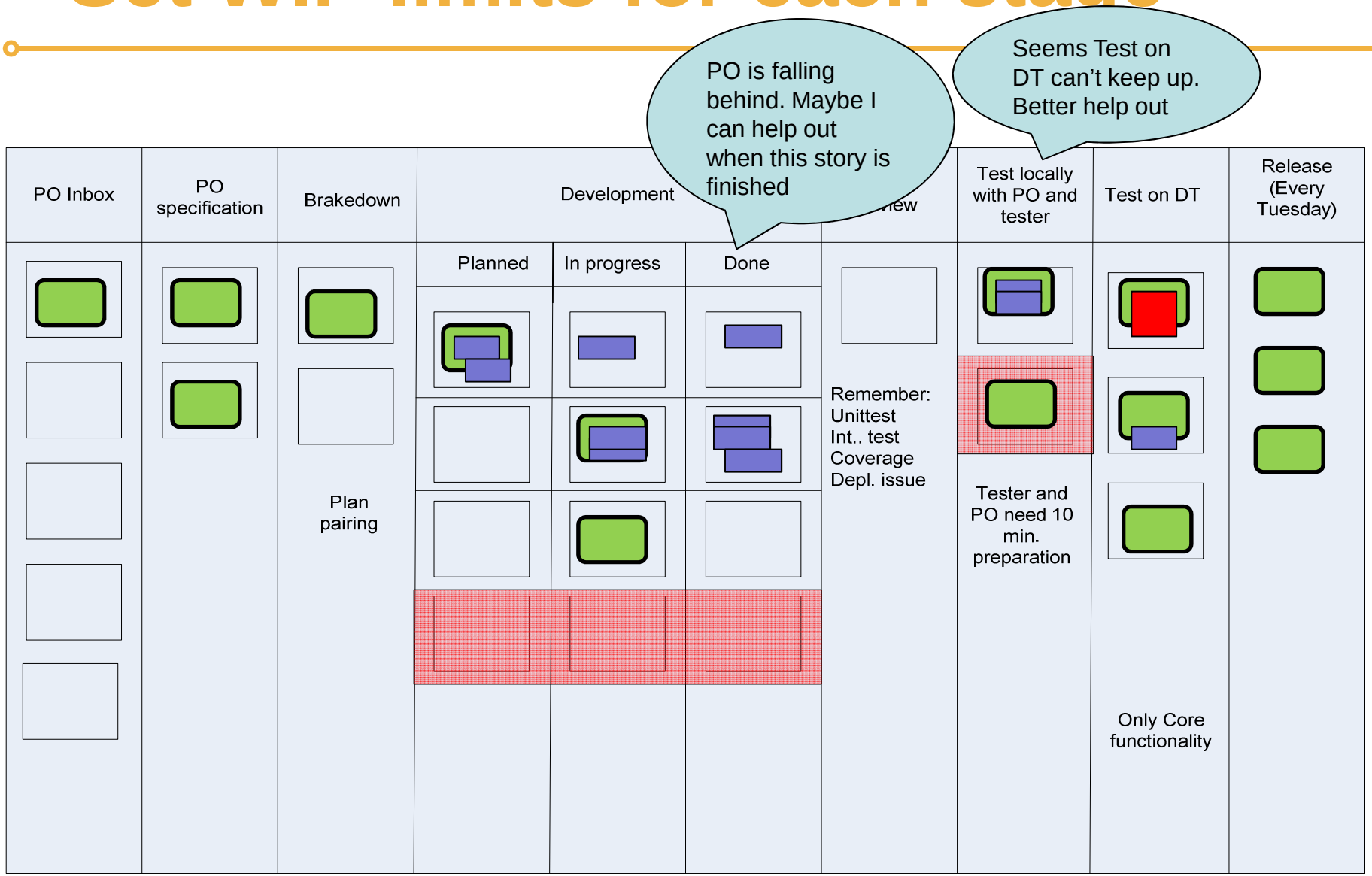
- Focus on business value and minimal marketable feature sets



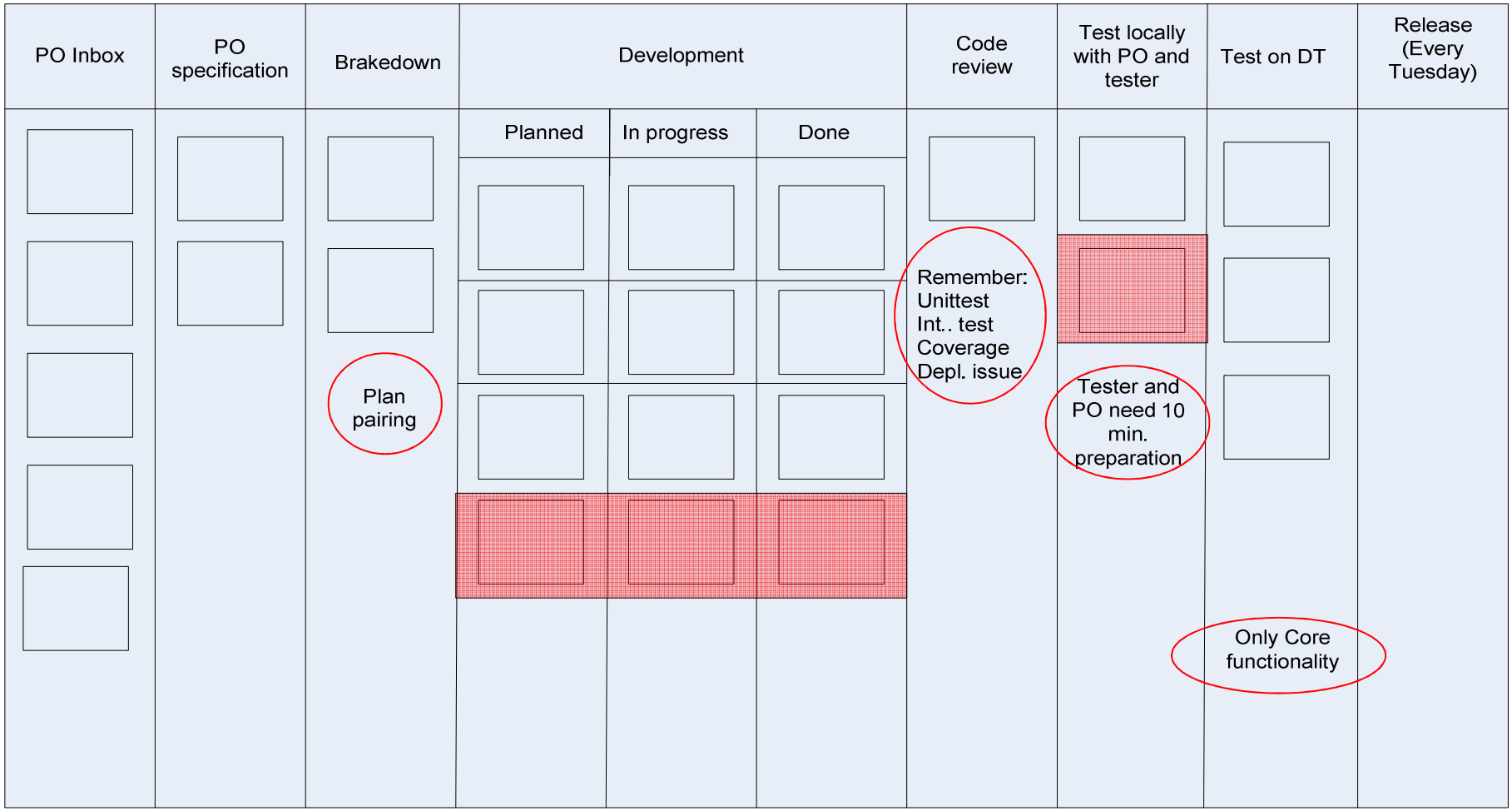
To achieve this

- Start by mapping the value stream and track work on a white board

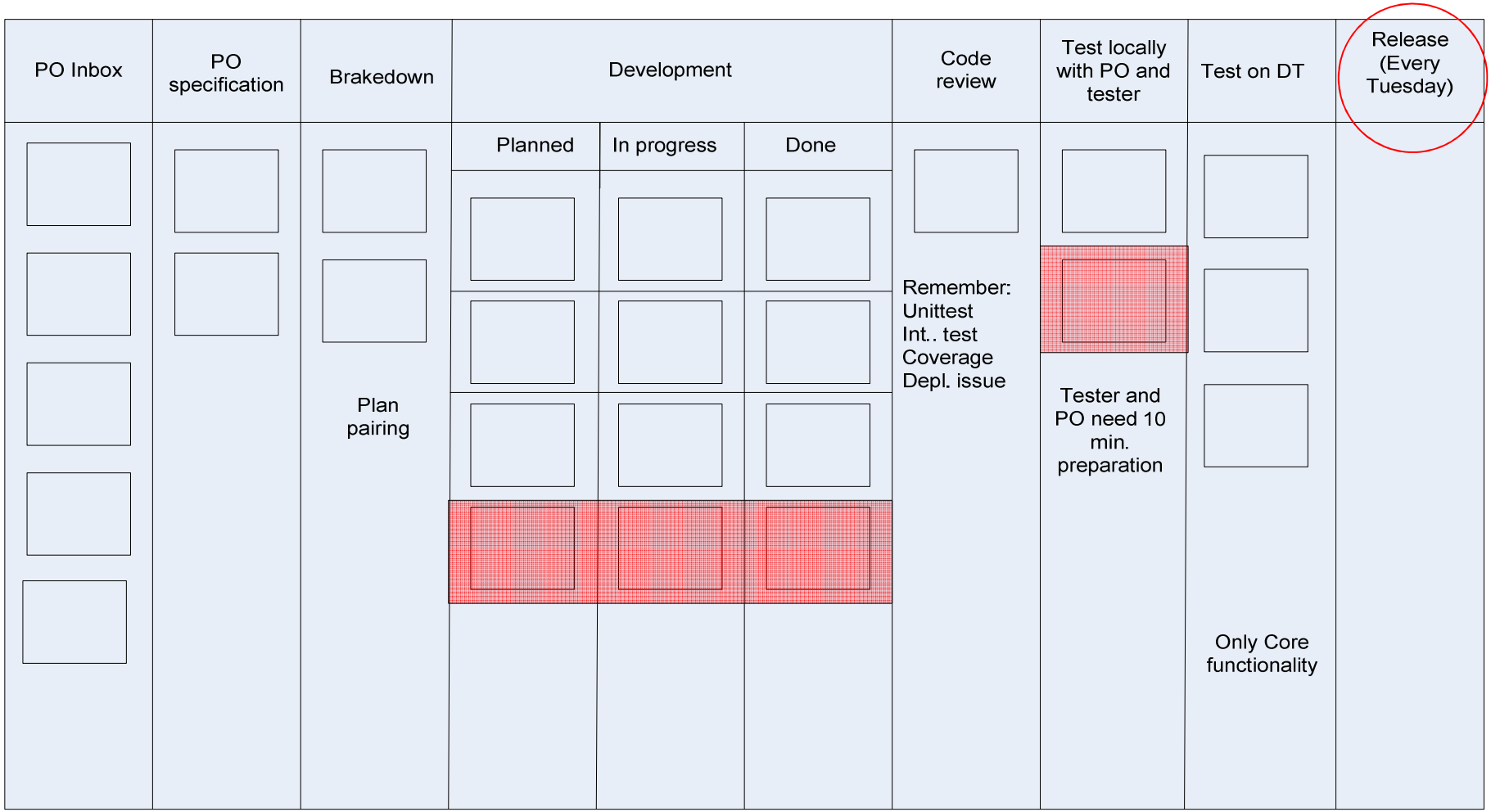
Set WIP limits for each stage



Use checklists to ensure quality



Release based on flow

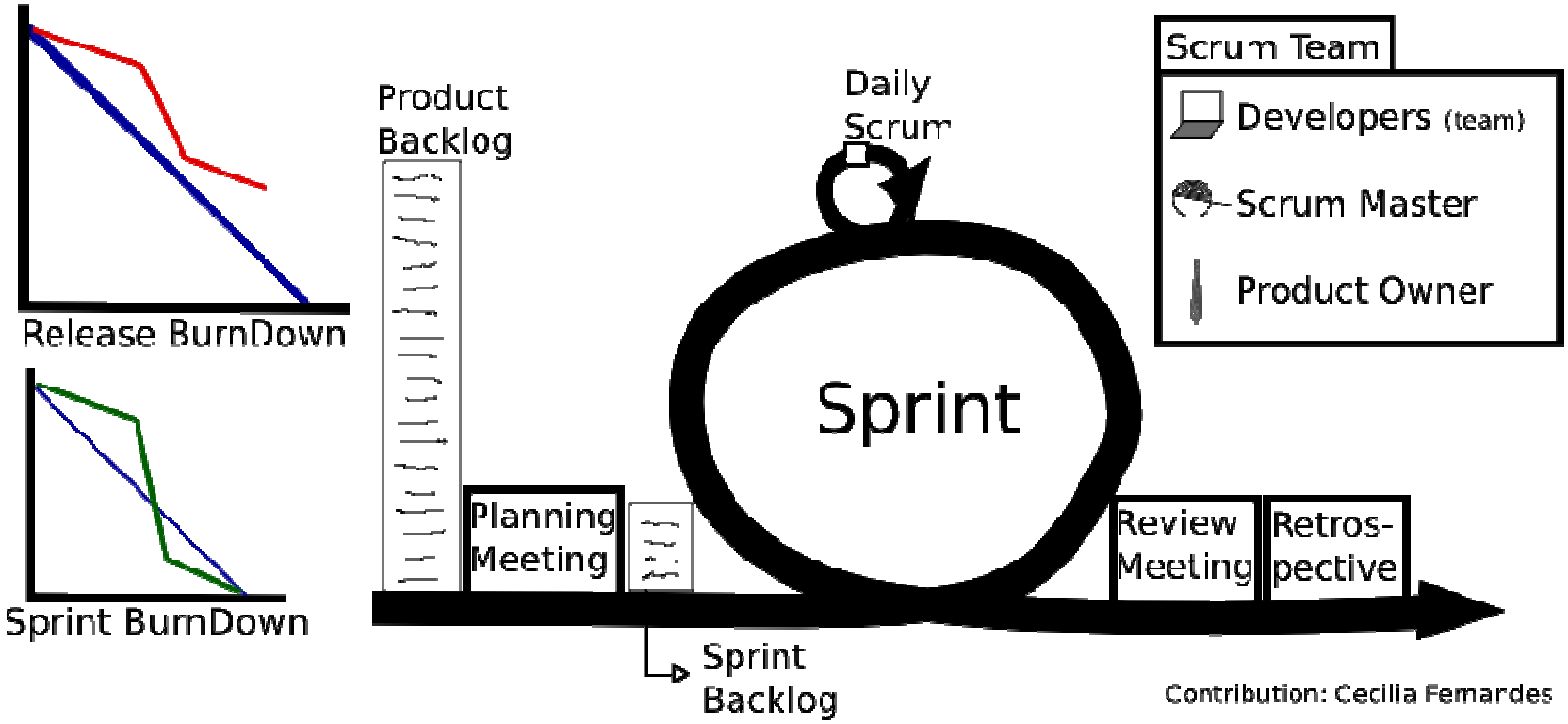


Pick the low hanging fruits

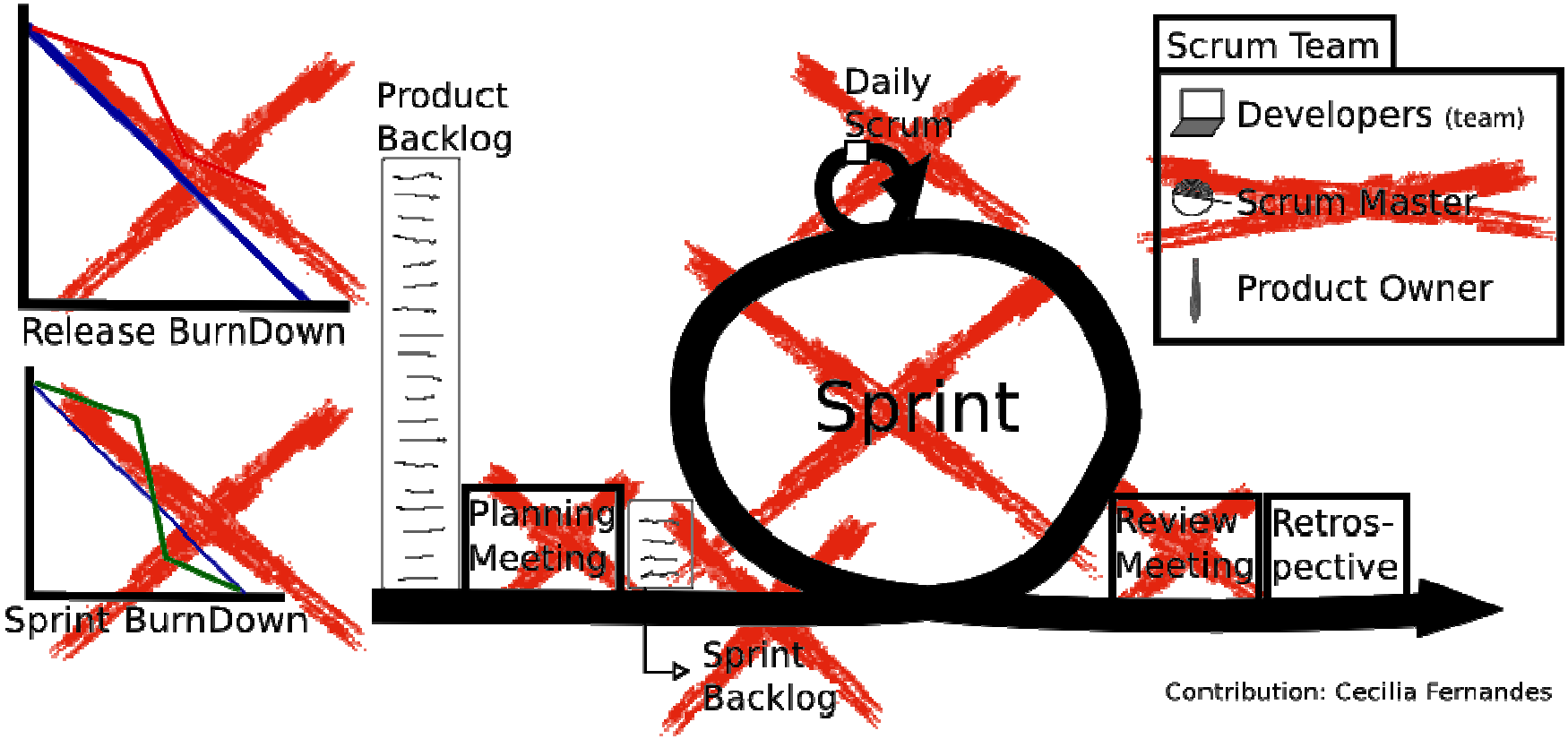
- You will be surprised how much you can achieve by
  - Mapping the value stream
  - Limiting work in progress.
  - Optimizing the whole

HOW DOES THAT FIT WITH CURRENT AGILE BEST PRACTICES?

You can do it


Release BurnDown
Sprint BurnDown
Product Backlog
Daily Scrum
Scrum Team
Developers (team)
Scrum Master
Product Owner
Sprint
Planning Meeting
Review Meeting
Retros-pective
Sprint Backlog
Contribution: Cecilia Femardes

You can drop it



Focusing on value sets instead of practices

- Using Kanban focus is no longer on specific practices
  - Choose practices that will help you use resources at hand most effectively in your context

But that is not my practice!!

David Anderson:

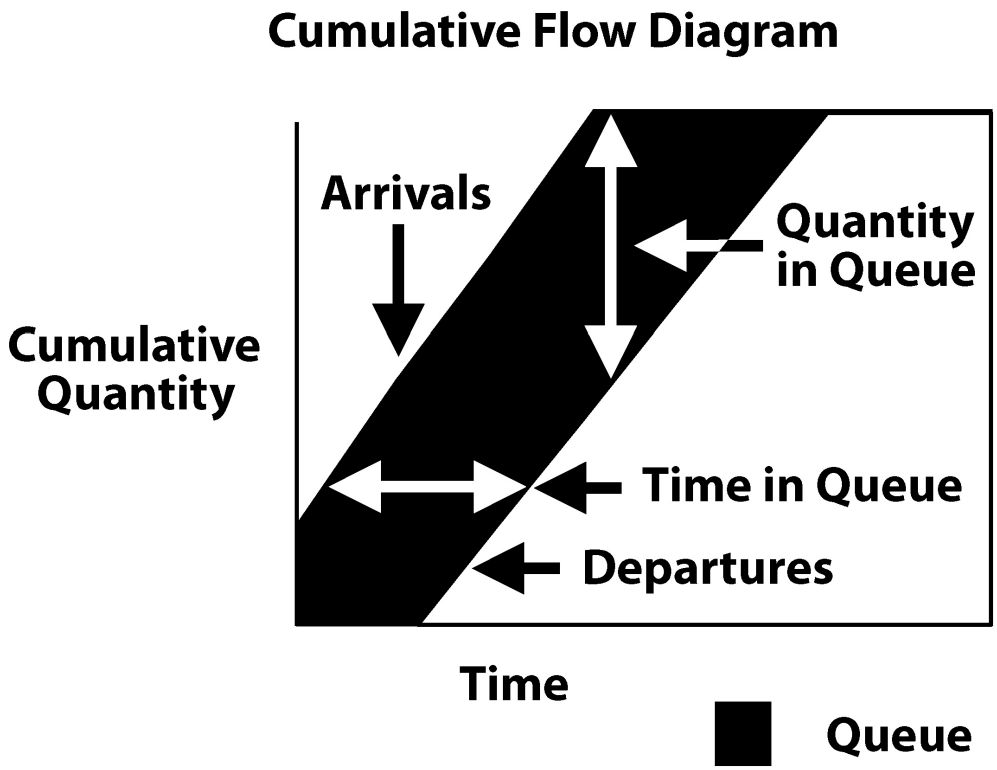
“I don’t care about your practices”

- Keep your eyes on the ball
  - We are hopefully using best practices because they deliver value

Loosing control?

- Kanban is NOT a “looser” way of doing Scrum
  - Metrics are just different

Cumulative Flow Diagrams



From "The Principles of Product Development Flow," by Donald G. Reinertsen.
Celeritas Publishing: 2009. Copyright 2009, Donald G. Reinertsen

To sum up

- Lean can help us understand the nature of Agile Anti-Patterns
- Kanban provides an excellent framework for applying Lean principles to software development

BUT THERE ARE NO FREE MEALS

Difficulties

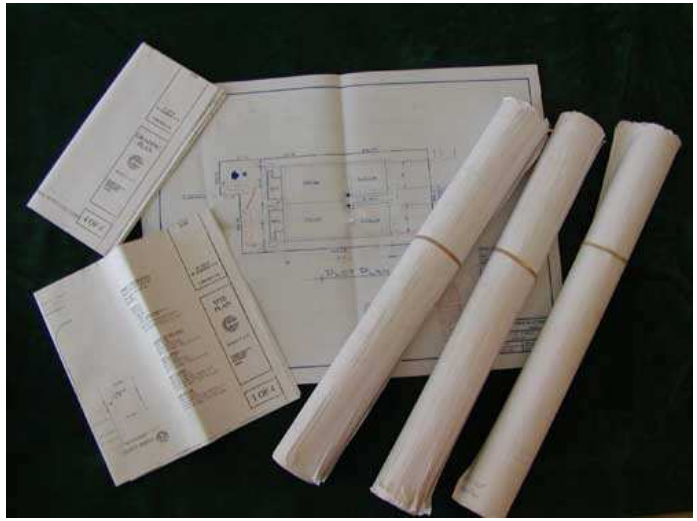
- People react very differently to the new structure
  - Some find it very hard to stay focused while others take on more responsibility and become true craftsmen

Difficulties

- Takes more effort to stay focused on releases

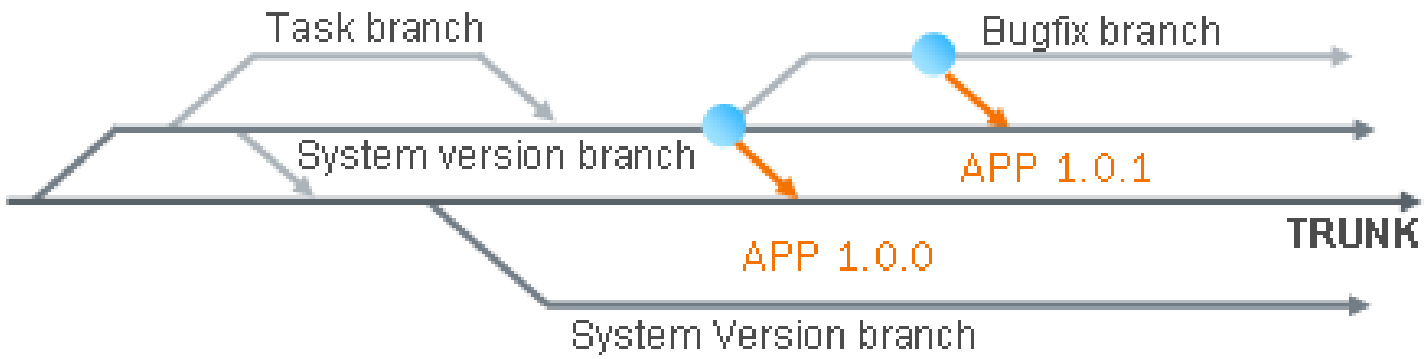
Difficulties

- Stronger need for overall plans and long term goals
  - Since people are no longer as focused on the short term goal



Difficulties

- Controlling continuous integration
  - When features are increasingly branched and merged to trunk to allow for fixed release dates without fixed scope



Difficulties

- Wrong perception of Lean

KANBAN EQUALS FEWER RESTRICTIONS

DOES THAT MEAN IT CAN ONLY BE USED BY MATURE TEAMS?

12 years old

had a dog

hit by a car

learned

restrictions are good

28 years old

a daughter

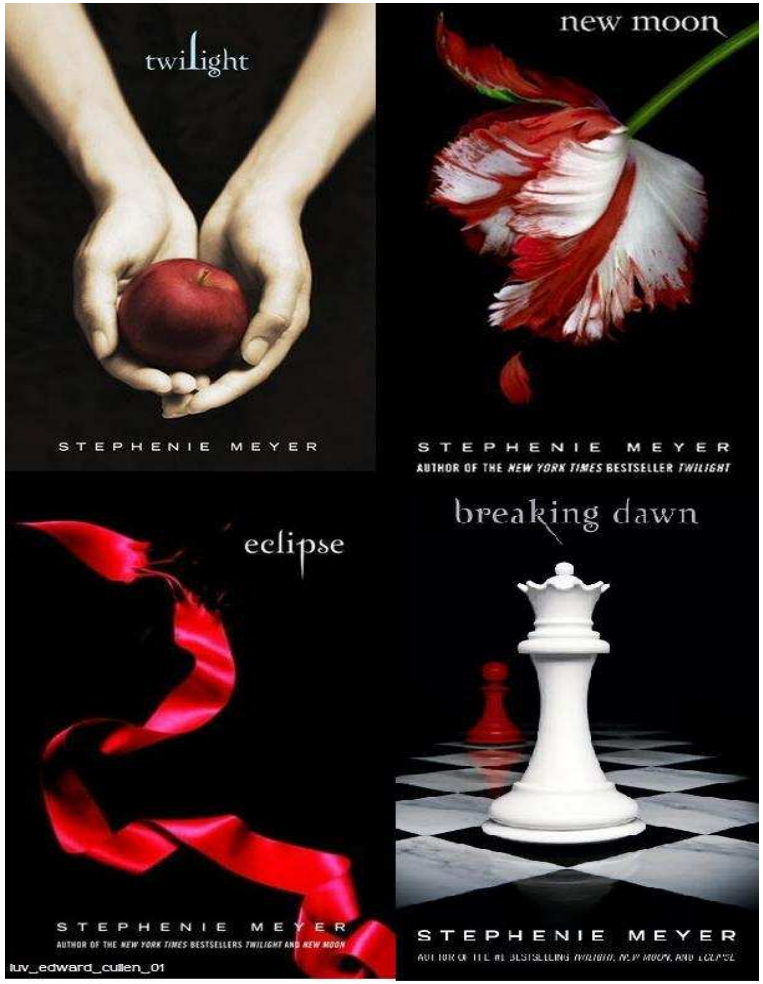
restrictions

although sometimes good

restrictions are also bad

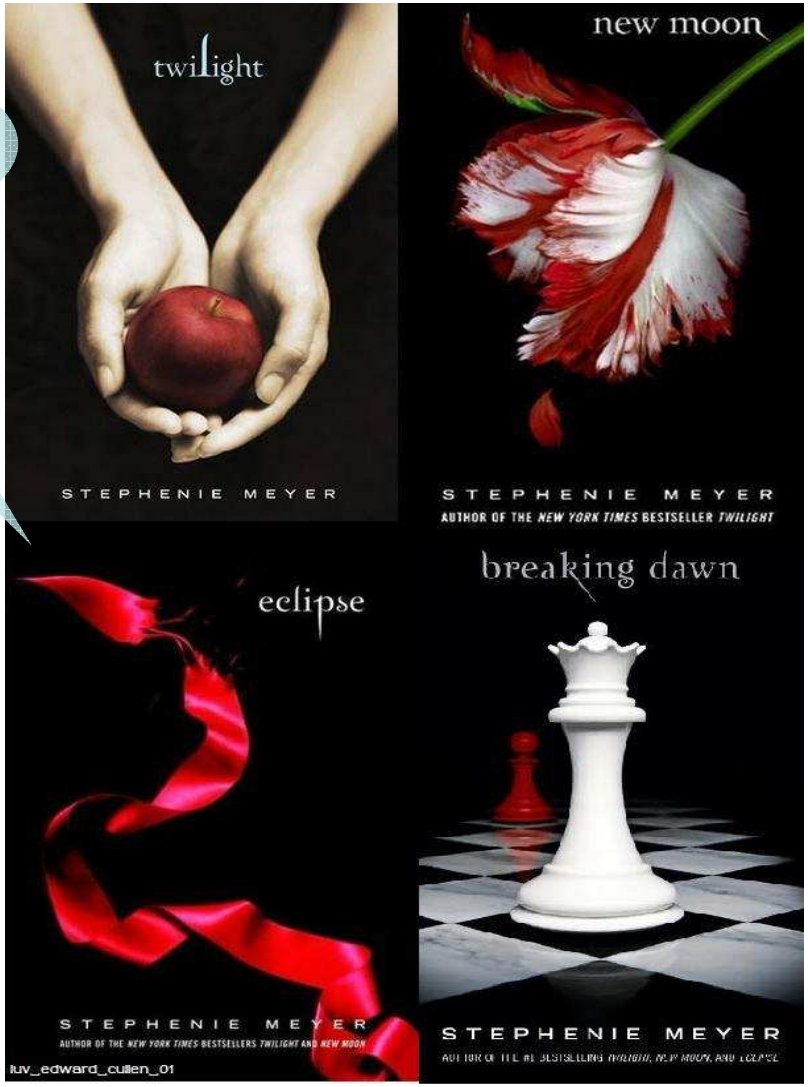
extra restrictions

recurring problems?



add restrictions

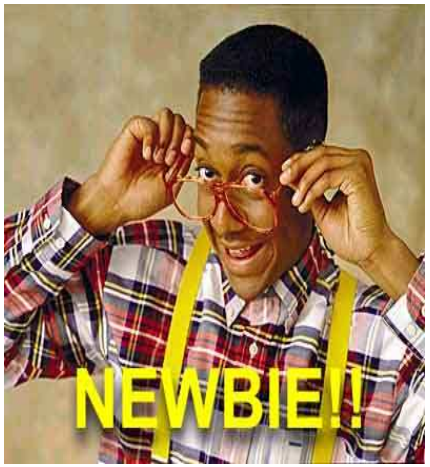
kill the vampire
and the werewolf



OR

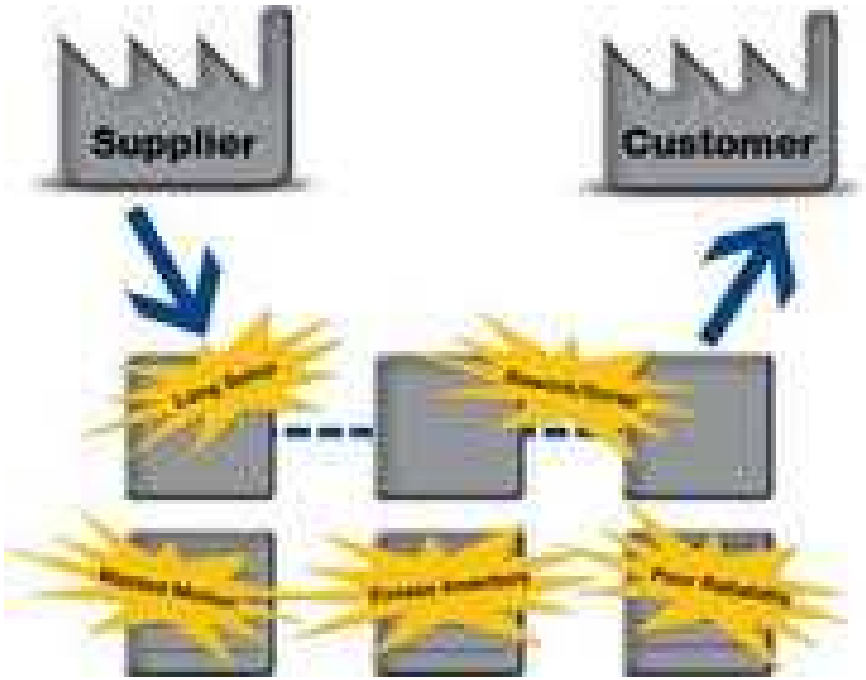
Kanban is a good way to start

- Since Kanban does not include specific practices you can start with your current process and improve it one step at a time



Step 1

- Visualize your current value chain



Step 2

- Implement/improve one practice at a time and gradually improve your process by focusing on flow, bottlenecks and limiting WIP

Why?

- Because many Agile projects fail because they want to do everything at once and faster than they or the organization is able to handle

QUESTIONS?

Contact information

- Jesper Boeg
  - Mail: jbo@trifork.com
  - Mobile: +45 51 54 28 20
  - Twitter: J_Boeg
- Guilherme Silveira
  - Mail: guilherme.silveira@caelum.com.br
  - Mobile: +55 11 83358084
  - Twitter: @guilhermecaelum

EXTRA

NOTES ON PLAN DRIVEN ITERATIONS

Plan driven iterations

- We are responsible for teaching our customers and ourselves
  - We will deliver exactly what we planned
  - The world is “Frozen” during the iteration
  - Business value should always fit a “2 week iteration”

Plan driven iterations

- From a Lean perspective iteration planning, test, deployment, equals - Batch production

Plan driven iterations

- Batch optimization is built on the faulty belief that processing big batches we can make the individual machine/fase go faster
  - Restricting flow
  - Increasing inventory
  - Reducing quality

Plan driven iterations

- “We can’t do 2 week iterations because of iteration review/planning overhead”
  - Shows you are still living in the old world of “Batch production” optimization
  - Instead focus on reducing transaction costs

Kanban is “Leaner” than traditional Agile Methods

- But remember to distinguish between Lean manufacturing and Lean Product Development
  - You cannot eliminate variability without eliminating value added in LPD
  - Cost of delay in manufacturing is often the same

Why I like fixed iteration length

- Lowers transaction costs
- Makes planning easier
- Facilitates continuous improvement

Look at the entire value stream

- Start by acknowledging that development is not always the bottleneck
- In cases where this is true you would rather want developers doing nothing than stressing the real bottleneck further
  - Ideally developers are of cause helping relieving the real bottleneck
- In traditional Agile methods, development is almost by definition regarded as the bottleneck
  - Keeps you from exposing the real bottleneck
  - Keeps you from taking the right actions do improve your process
  - It took a switch from Scrum to Kanban for us to realize this

Kanban is just a process

- Sometimes one process will work better than another and sometimes they will be equally good.
  - Understand your problem before trying to solve it.
  - Expand your toolkit.
  - My tool is better than yours attitude won’t get you anywhere
  - Compare processes to understand them not for judgment.

Kanban is just a process

- You NEED good practices
  - Agile product management principles do not work well without good practices to support them
  - Quality built in is not just well tested. It is also good architecture and good coding practices

Practices

- If you haven‘t got the technical practices in place it doesn’t matter what process you are using,
  - It won’t get you anywhere in the long run.
  - But a good process will help you focus on having good technical practices – and I will argue that Kanban does that exceptionally well.

Ask yourself

- Are you environment driven or environment driving?
  - Methods
  - Organization
  - People
  - Technology
- That could very well be your biggest impediment since it stops continuous improvement

Look at your process from a true Lean perspective

- Don’t try to make a process seem Lean just because it’s a popular word
- A team pulling items from a backlog does not make it a pull system
  - It only means that you have a pull mechanism within your system
  - It doesn’t keep you from delivering more functionality than the customer needs or is able to adopt.
- A true pull system is based on the entire value stream and making sure it is closely aligned with the needs and capabilities of the customer
  - A software Kanban system should represent such value stream