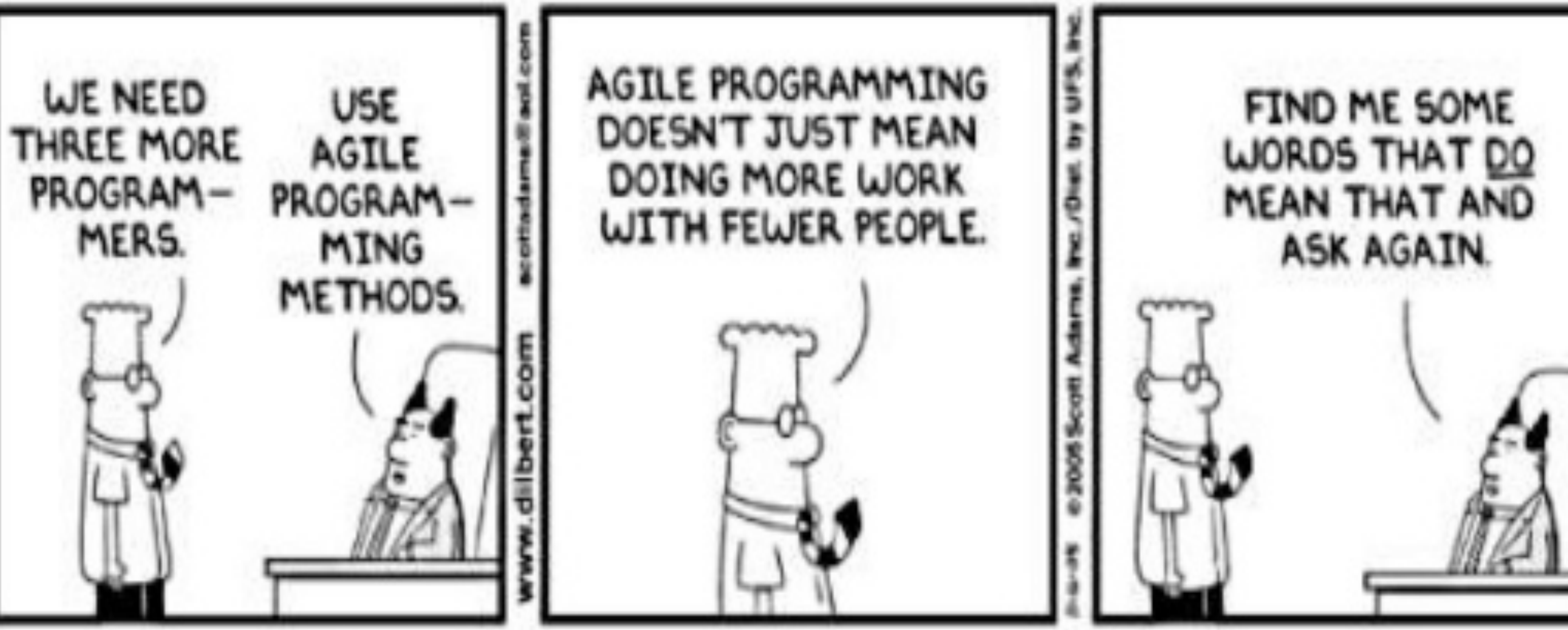
SCRUMBAN AND A3 THINKING DEMYSTIFIED

Ajay Reddy

ajay@codegenesys.com | @ajrdy

CodeGene5ys

What Scrum and Scrumban are not


WE NEED THREE MORE PROGRAM-MERS.
USE AGILE PROGRAM-MING METHODS.
www.dilbert.com
scottadams@aol.com
AGILE PROGRAMMING DOESN'T JUST MEAN DOING MORE WORK WITH FEWER PEOPLE.
© 2006 Scott Adams, Inc./Dist. by UFS, Inc.
FIND ME SOME WORDS THAT DO MEAN THAT AND ASK AGAIN.

© Scott Adams, Inc./Dist. by UFS, Inc.

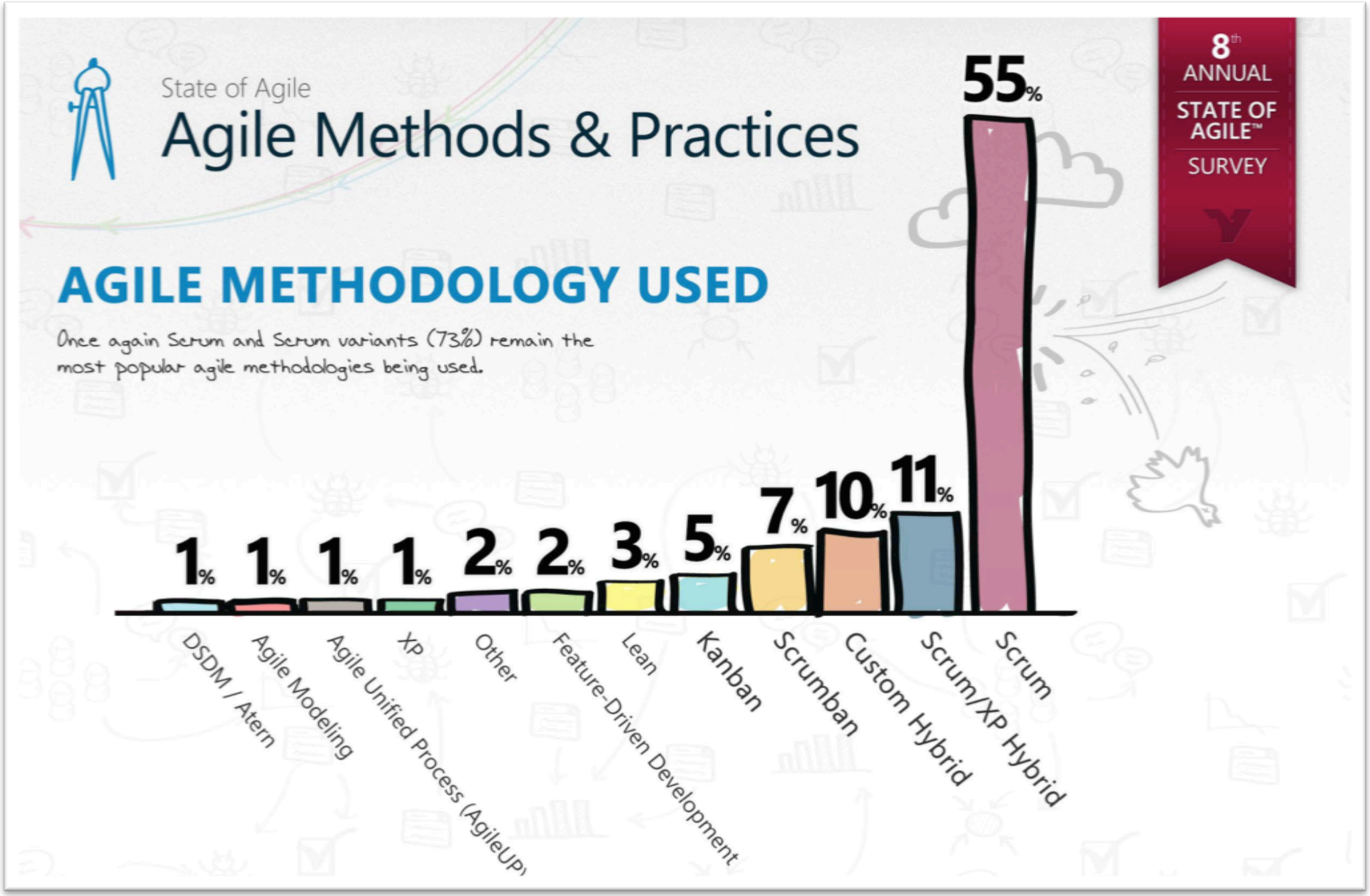
Topics

- Why and What Scrumban
- Scrumban Foundations and Evolutions at Mammoth Bank
- Scrumban measurements (Optional)

Section 1 of 3

Why and what Scrumban

VersionOne's State of Agile across the industry



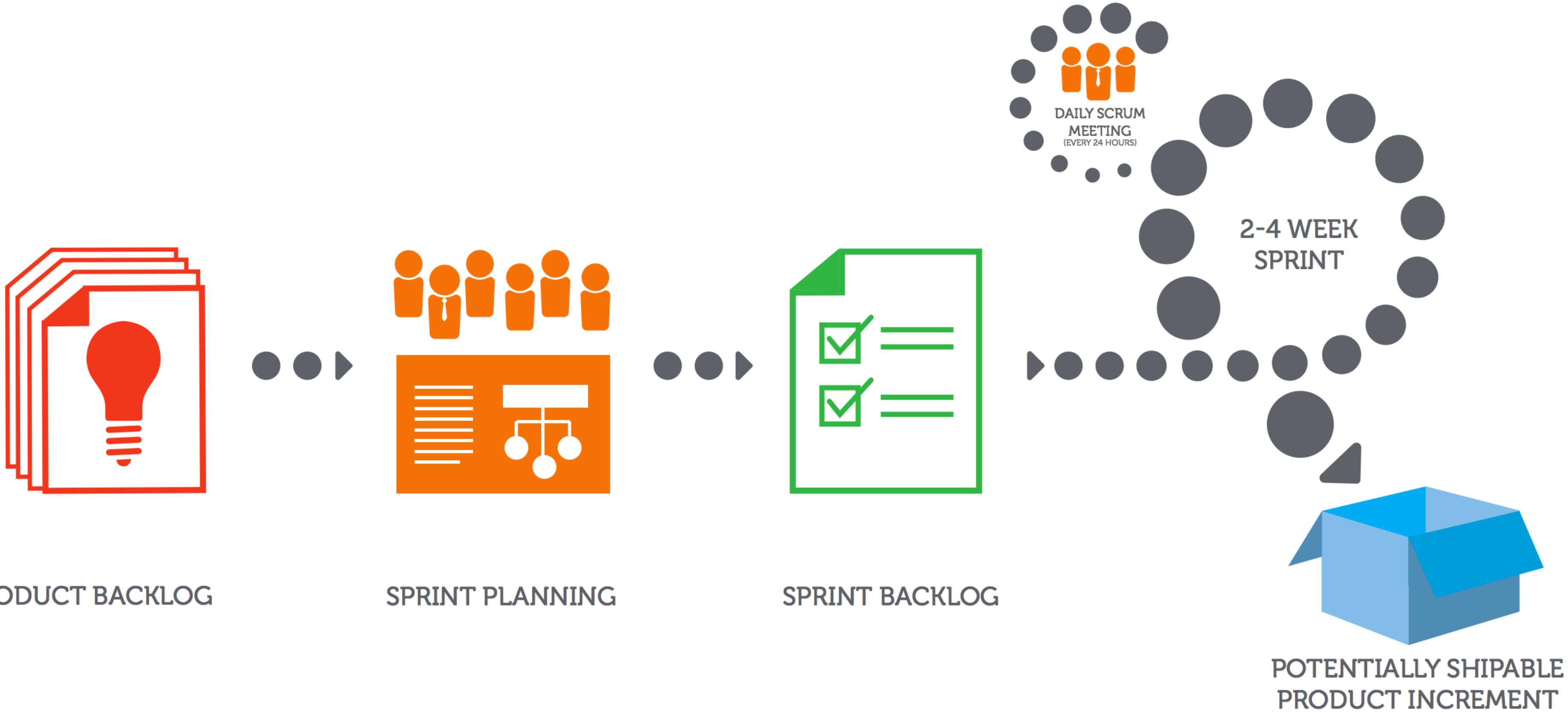
State of Agile

Agile Methods & Practices

AGILE METHODOLOGY USED

Once again Scrum and Scrum variants (73%) remain the most popular agile methodologies being used.

Scrum



Source: Scrum Alliance

> “75% of organizations using Scrum will not succeed in getting the benefits that they hope for from it.”

-Ken Schwaber

ScrumbanDO

Why Scrumban

- managing variability in knowledge
- Improves predictability
- imposes no prescribed process destination
- provides more natural and contagious mechanisms for scaling improvements across an enterprise
- can germinate from anywhere within an organization
- product / organization management features into the mix such as risk, value & cost.

Mammoth Bank

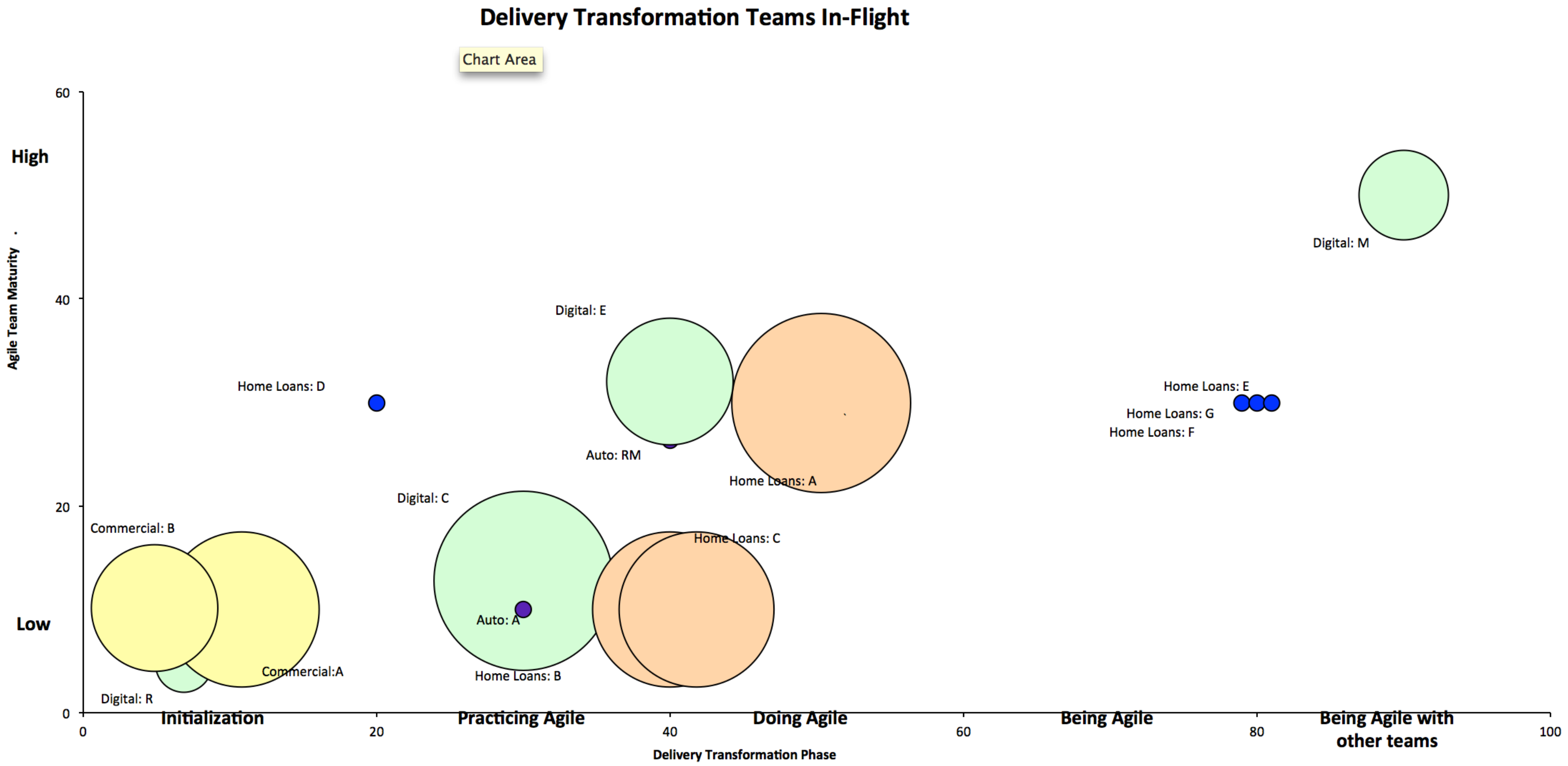
Mammoth Bank is a large national (international) bank headquartered somewhere in the US.

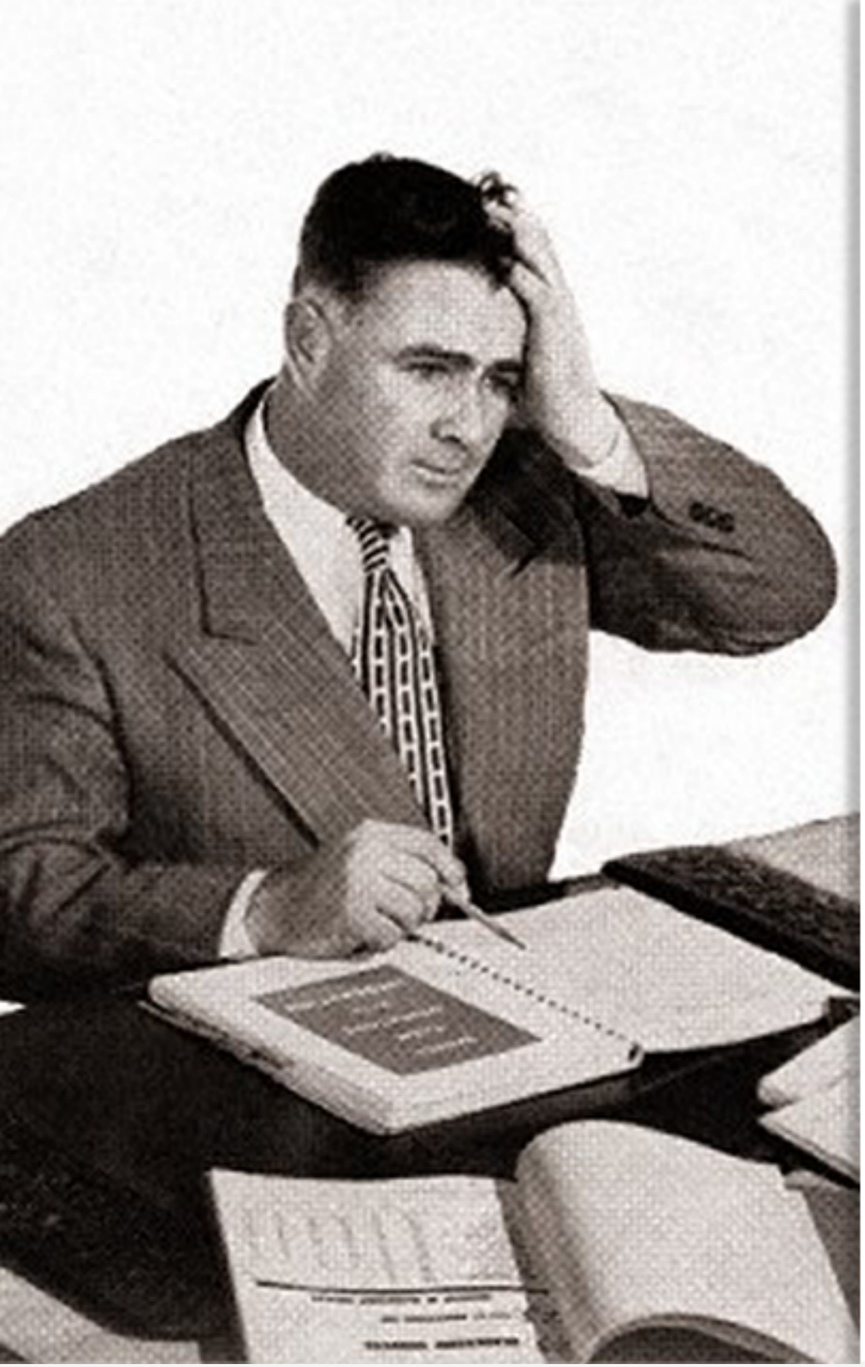
It has a very large IT organization (several thousand strong) spread across multiple locations, including several offshore partners.

History and current state of Agile transformation

- 3 revision of Agile transformation
- In 2013, the bank was faced with several difficulties:
  - Disruptive change not working
  - Premature Success declared
  - Coaching with inadequate results
  - Some improvements clearly present

Vague and False impressions



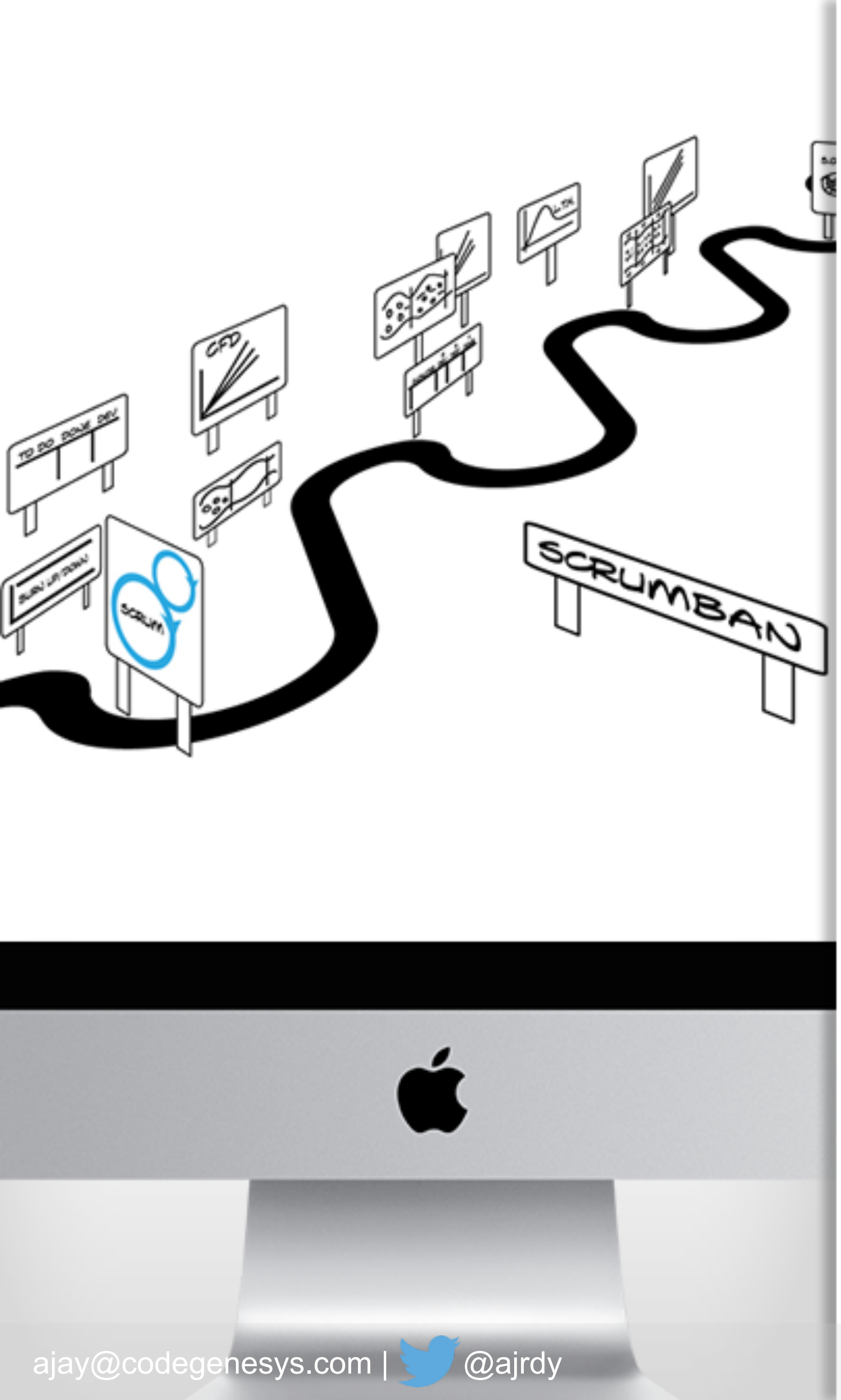


Bank’s problem areas

- Productivity
- Predictability
- Politics
- Profitability
- Permanence
- Plasticized
- Prematurity

Bank's goals

- greater flexibility (agility)
- improved predictability (dependability)
- increased productivity
- higher quality
- better alignment
- greater customer satisfaction
- GREATER MARKET SHARE
- SUSTAINABLE PROFITABILITY



Scrumban Definition

Scrumban is the Kanban method applied to Scrum as the starting and underlying process framework.

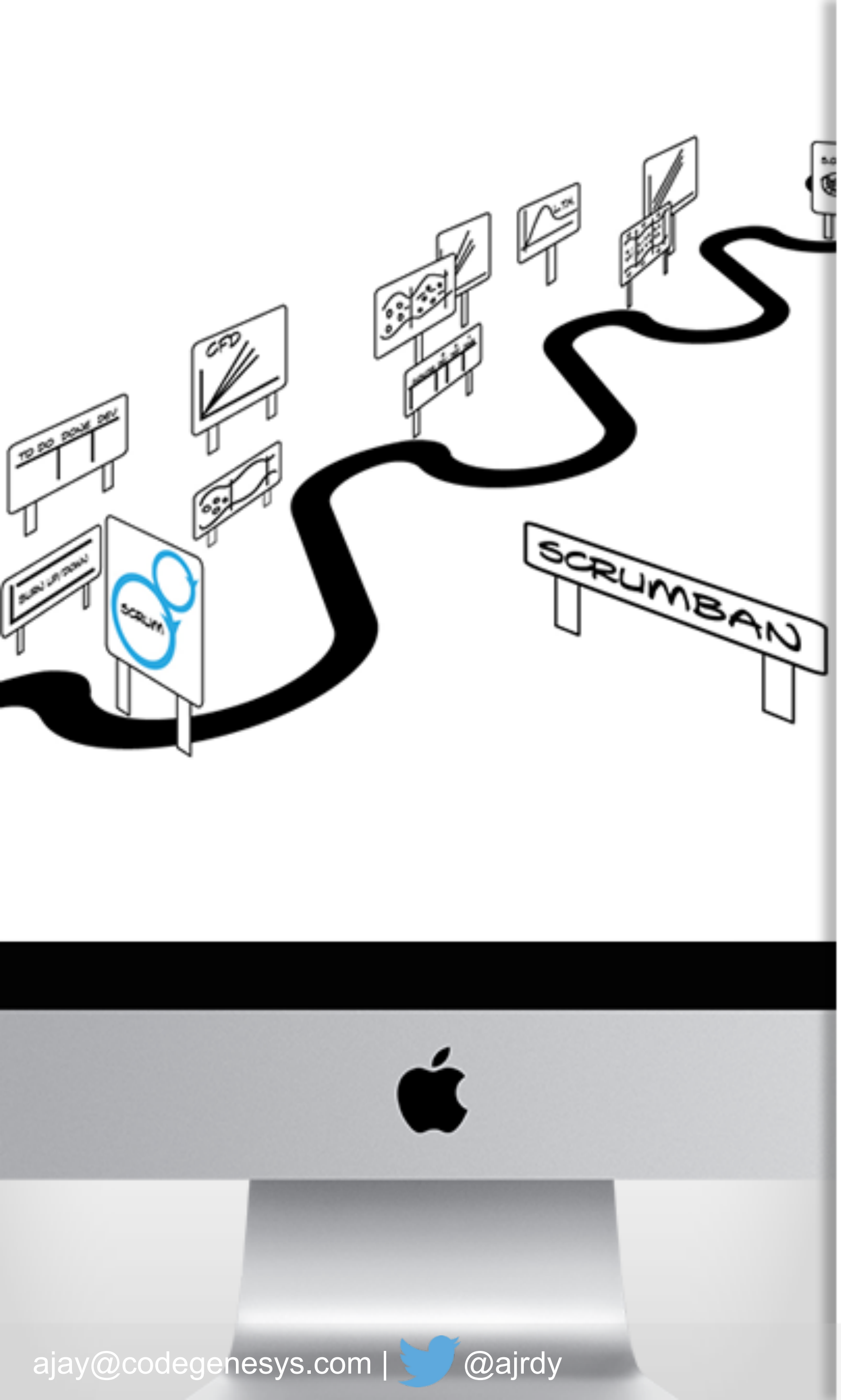
The Kanban Method

4 Principles

- Start with what you do now
- Agree to pursue incremental, evolutionary change as opportunities are discovered
- Respect current process, roles, responsibilities and titles
- Encourage acts of leadership at all levels of the organization

6 Practices

- Visualize
- Limit WIP (‘kanban’)
- Manage flow
- Make policies explicit
- Develop feedback mechanisms at the workflow, inter-workflow and organizational levels
- Improve collaboratively, evolve experimentally (using models and the scientific method)



Scrumban Definition

Scrumban is the Kanban method applied to Scrum as the starting and underlying process framework.

Scrumban is a **pragmatic** and **entrepreneurial** application of The Kanban Method to Scrum, using scientific **management theories** often resulting in a **continuing evolution** of Scrum based systems into business valued, service oriented, Agile, Lean and adaptive systems at a **pace moderated** by the thinking systems.

Scrumban

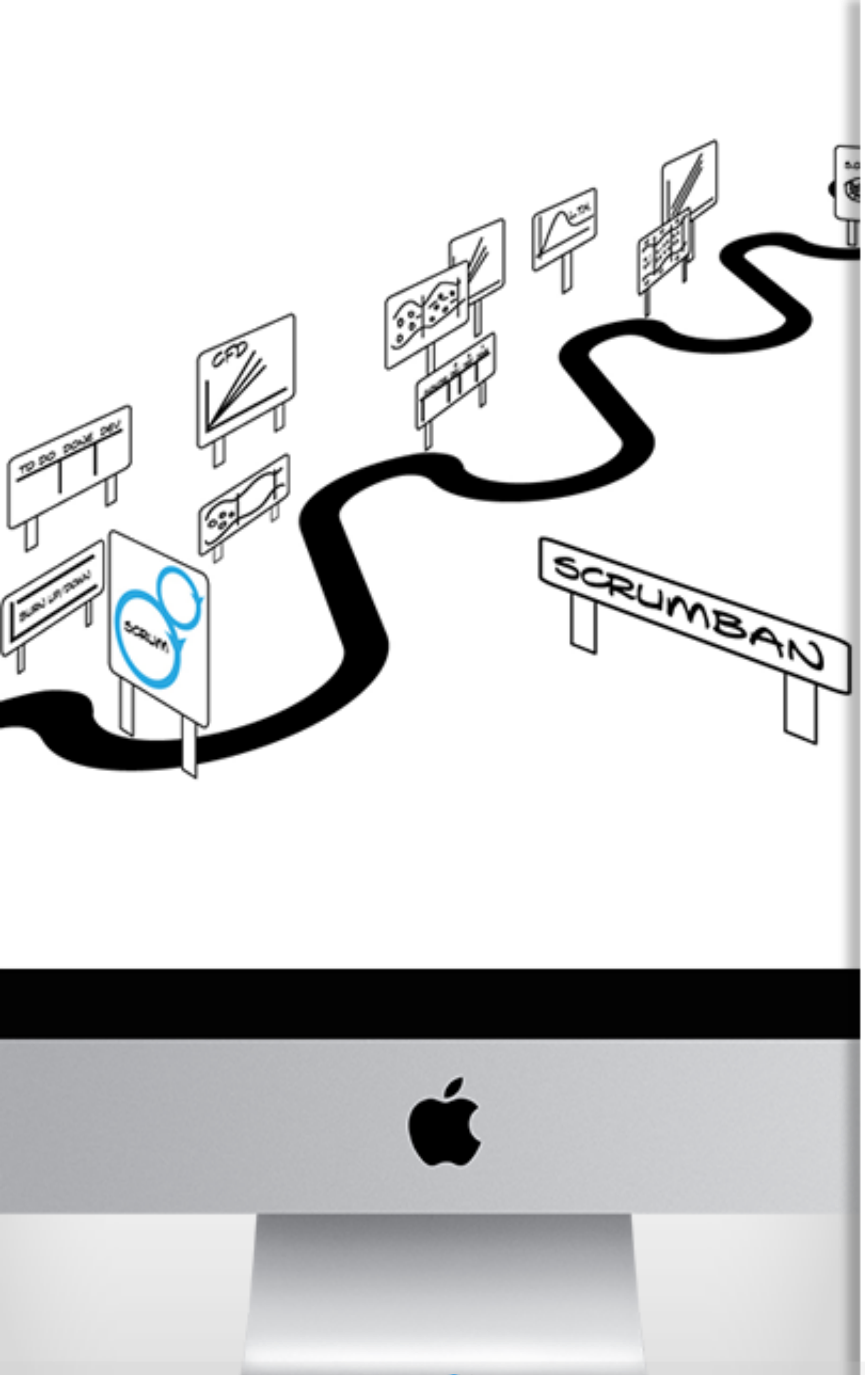
False Hybrids and common 'macro' evolutions

"Scrum in Kanban"

"Kanban in Scrum"

Scrumban is not a process process framework or a process destination!

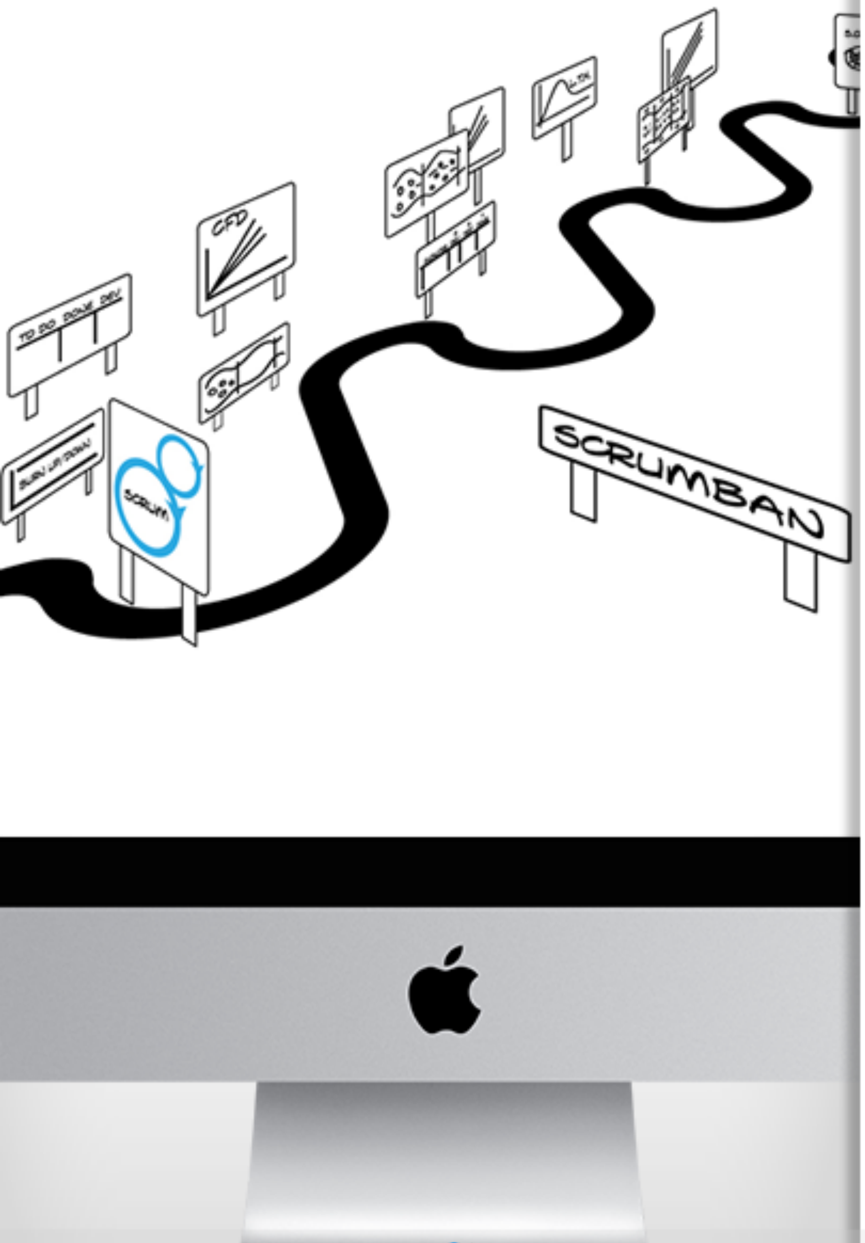
Scrum's iteration/sprint is an instance of Kanban's synchronized two phased commit.



Scrumban Definition

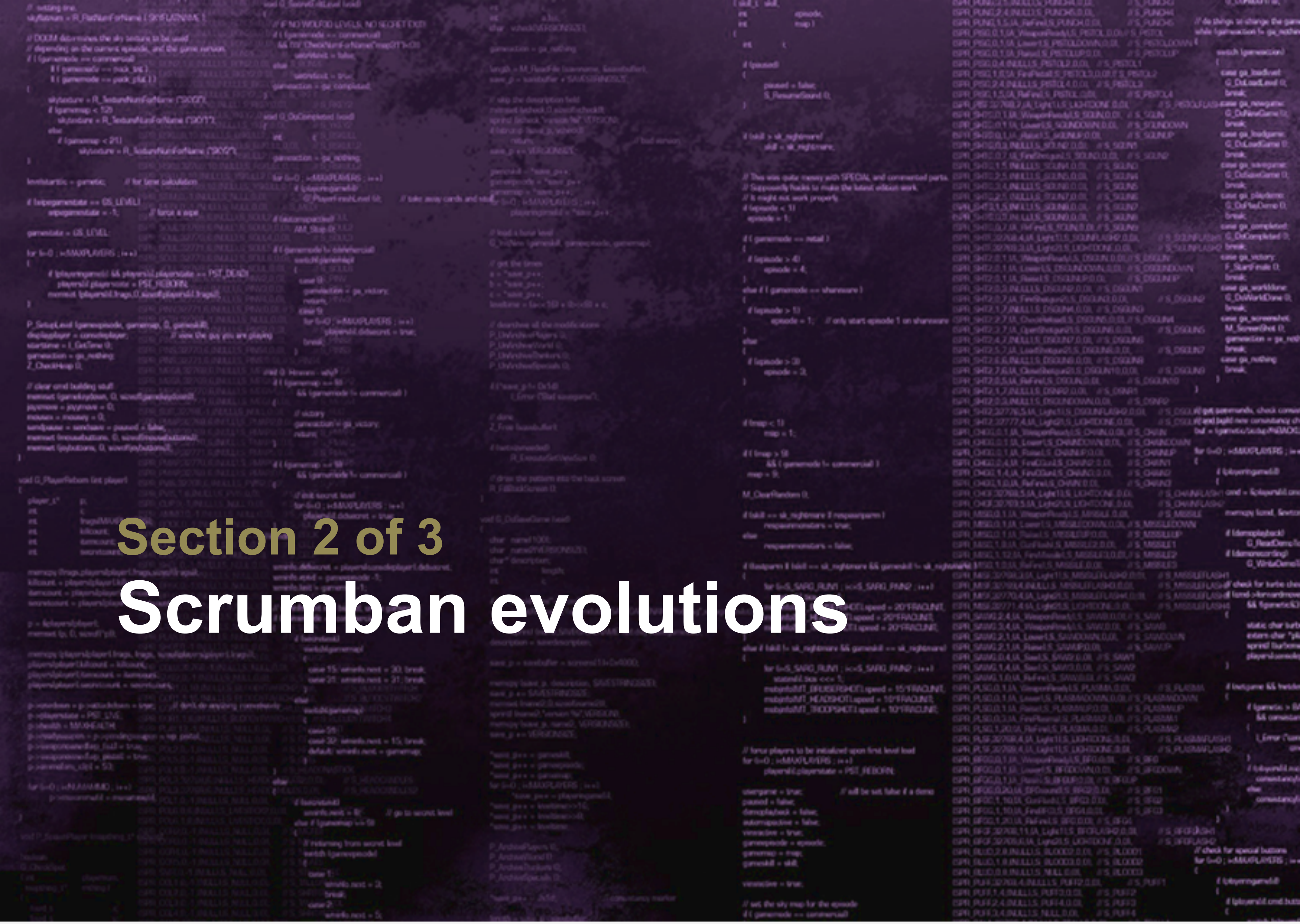
Scrumban is the Kanban method applied to Scrum as the starting and underlying process framework.

Scrumban is a **pragmatic** and **entrepreneurial** application of The Kanban Method to Scrum, using scientific **management theories** often resulting in a **continuing evolution** of Scrum based systems into business valued, service oriented, Agile, Lean and adaptive systems at a **pace moderated** by the thinking systems.



Some common examples of change (not comprehensive):

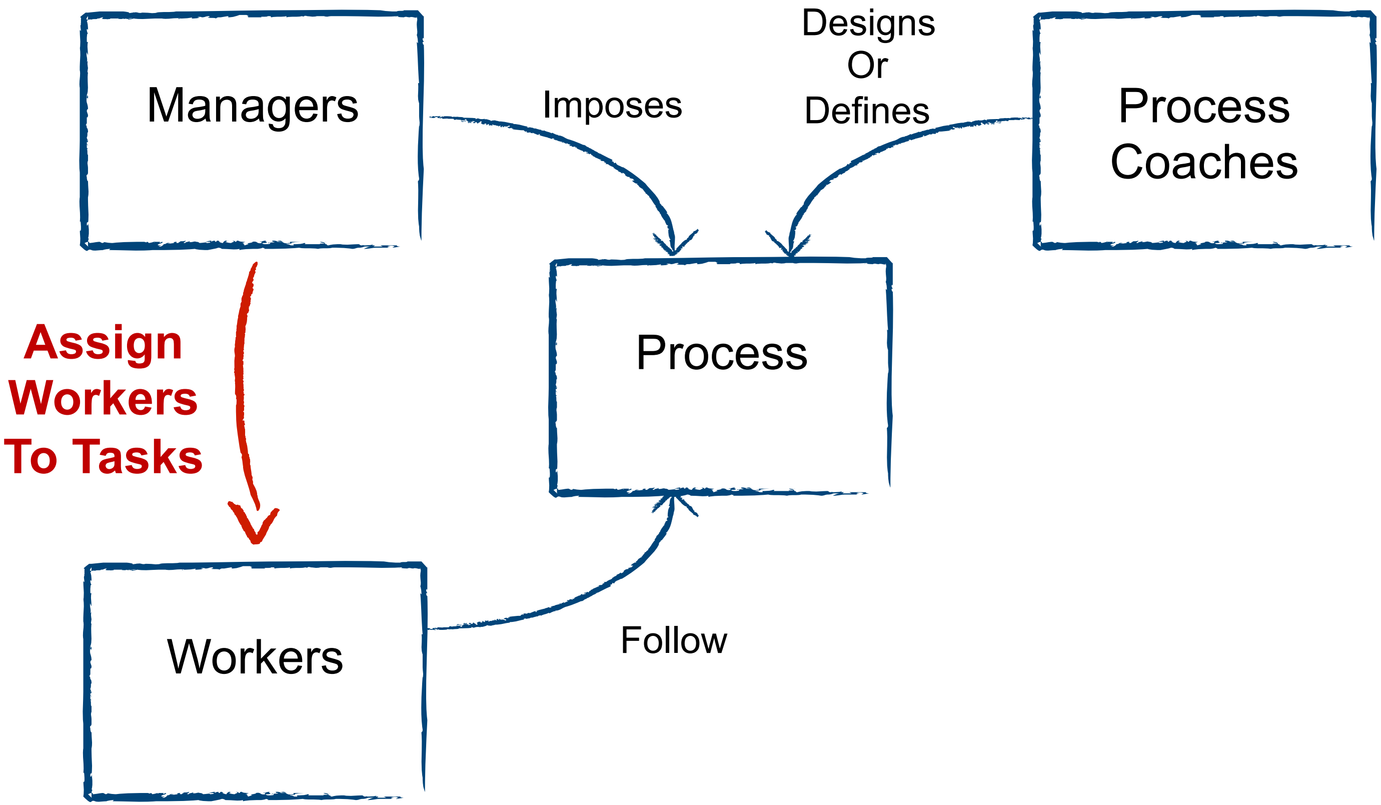
- Flow within an iteration (Two phased synchronized commit)
- Iteration planning
- Prioritization
- Role definitions
- MVP/MMF and Release planning
- Spikes
- Story points (not official part of Scrum)
- Budgeting
- Epics and story break down



Section 2 of 3

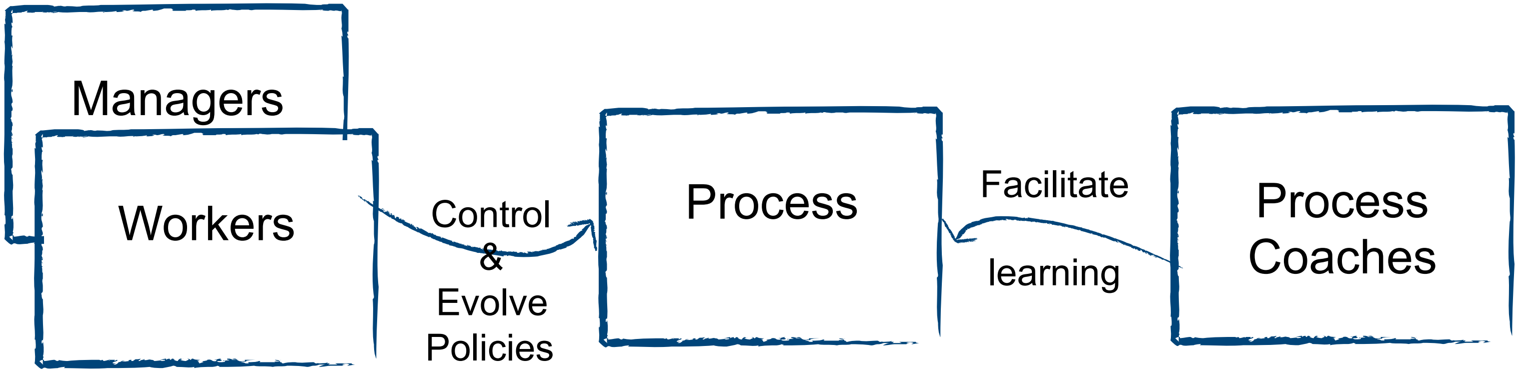
Scrumban evolutions

The Old Model (18th-20th Century)



Source: David Anderson, Agile Russia

The New Model (21st Century)



Source: David Anderson, Agile Russia

RAD Group

Significantly delayed development of a consumer-facing mobile application.

Bank was beginning to rapidly lose customers to other institutions.

5 separate team of 6-7 members each

Sam's Team

Starting Conditions

- Product Back log/Iteration back log/User stories
- Sprints Burn downs and Burn ups
- Scrum master, product owner, Agile coach
- Push model- assignments
- Consensus based points estimation
- Priority determined by product owner

Predictability/Reliability
Business effectiveness of results

Learning Evolutions

Mastery Level
A3 Thinking
Cynefin
Systems thinking
Long-Tail

Mobilize

- The Kick start framework
- Sources of dissatisfaction and Defining Purpose & Success Criteria
- Who are our customers
- Demand and capability Analysis
- Basic Management

Moblize

Already Familiar to Scrum Teams

- work items / work cards (user stories)
- work size estimate (story points)
- definition of done
- daily standups

New Concepts

- work type
- workflow
- pull
- ready / buffer columns
- blockers
- classes of service
- capacity allocations (explicit WIP limits) [Proto kanban]

System 1 (Sam's Team)

Product Backlog	Iteration Commitment	To do	Doing	Release	Done
🟪 🟨 DoD		Source Destination			

🟨 - Standard

🟪 - Bug Fix

Identified

Recognition of need for understanding demand and capability
What is the nature of demand from them?
Definition of done

Learning Evolutions

Mastery Level 🧍
A3 Thinking
Cynefin
Systems thinking 🧍
Long-Tail Priorities

Systems Thinking

Systems Thinking

- Dynamic systems
- Amplifying or Dampening
- Causal effects, Cascading effects
- Changing systems
- Skepticism
- Complexity within Lateral Thinking
- Lean toward the weightier understanding of the collecting of the system

System 1 (Sam’s Team)

Iteration Backlog	Analysis	Develop	Test	Deploy	Done
🟪 🟨 🟨 🟨					
	DoD	DoD	DoD		

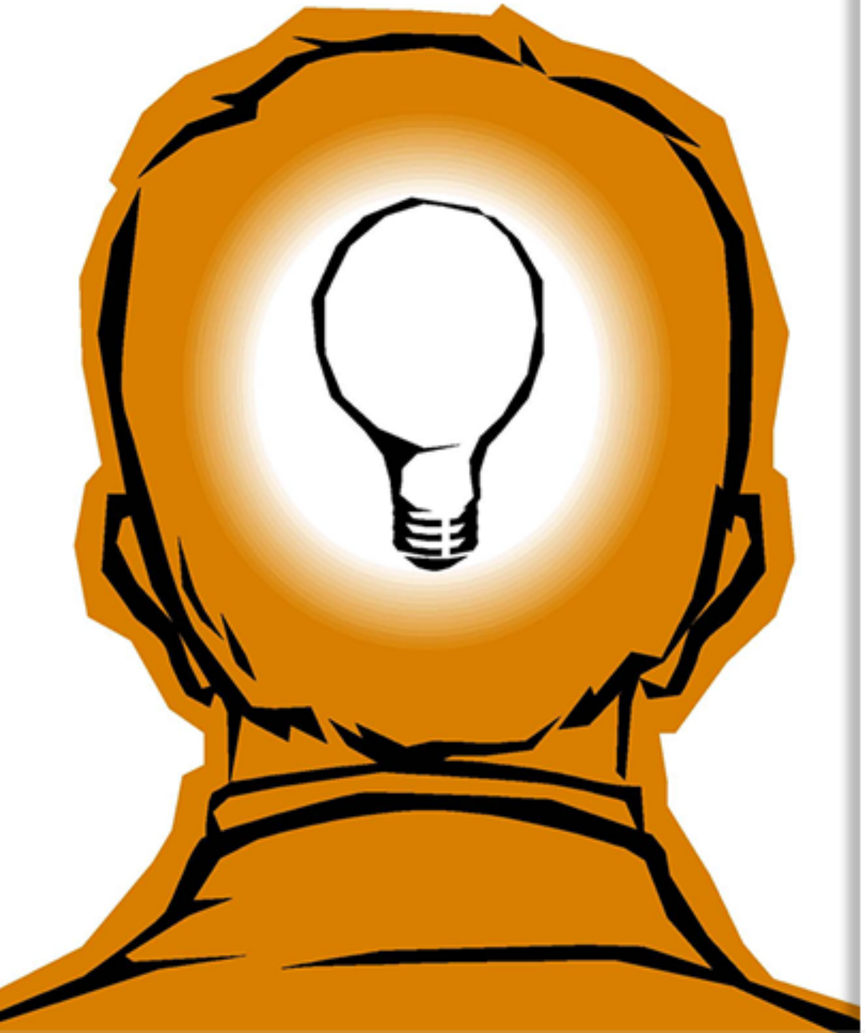
🟨 - Standard

🟪 - Bug Fix

How works is done
Customers Identified

Learning Evolutions

Mastery Level 🧍
A3 Thinking
Cynefin
Systems thinking 🧍
Long-Tail



Thinking Systems

Creates an adaptive capability in your organization

Enables adaptability in your business processes to respond successfully to changes in your business environment

Toyota's culture and adaptive capability's outcome

Toyota's quality circles

- Optional
- Collaborative
- Focused
- Empirical
- Common method

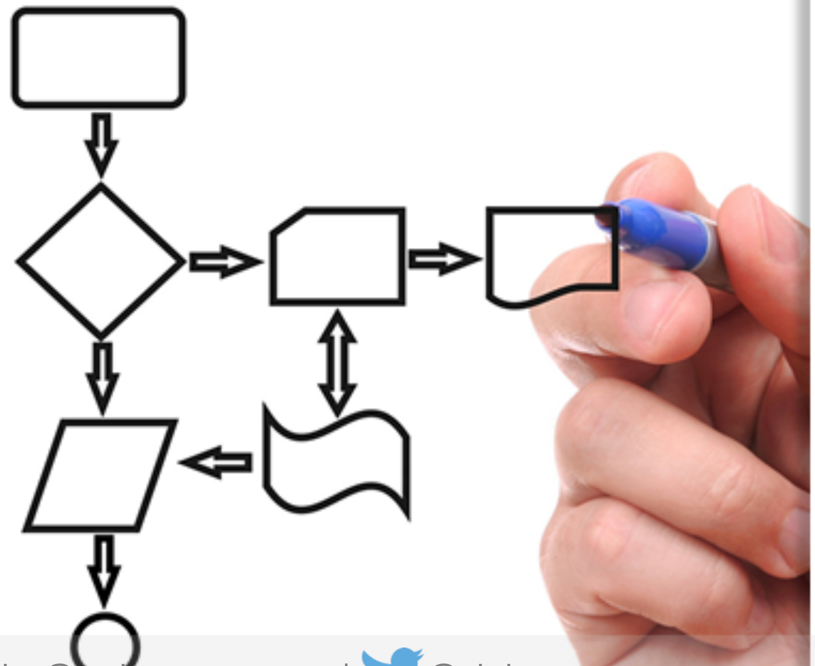
Logical

Objective

Means and ends important

Synthesis and presentation

Coherence-Continuity



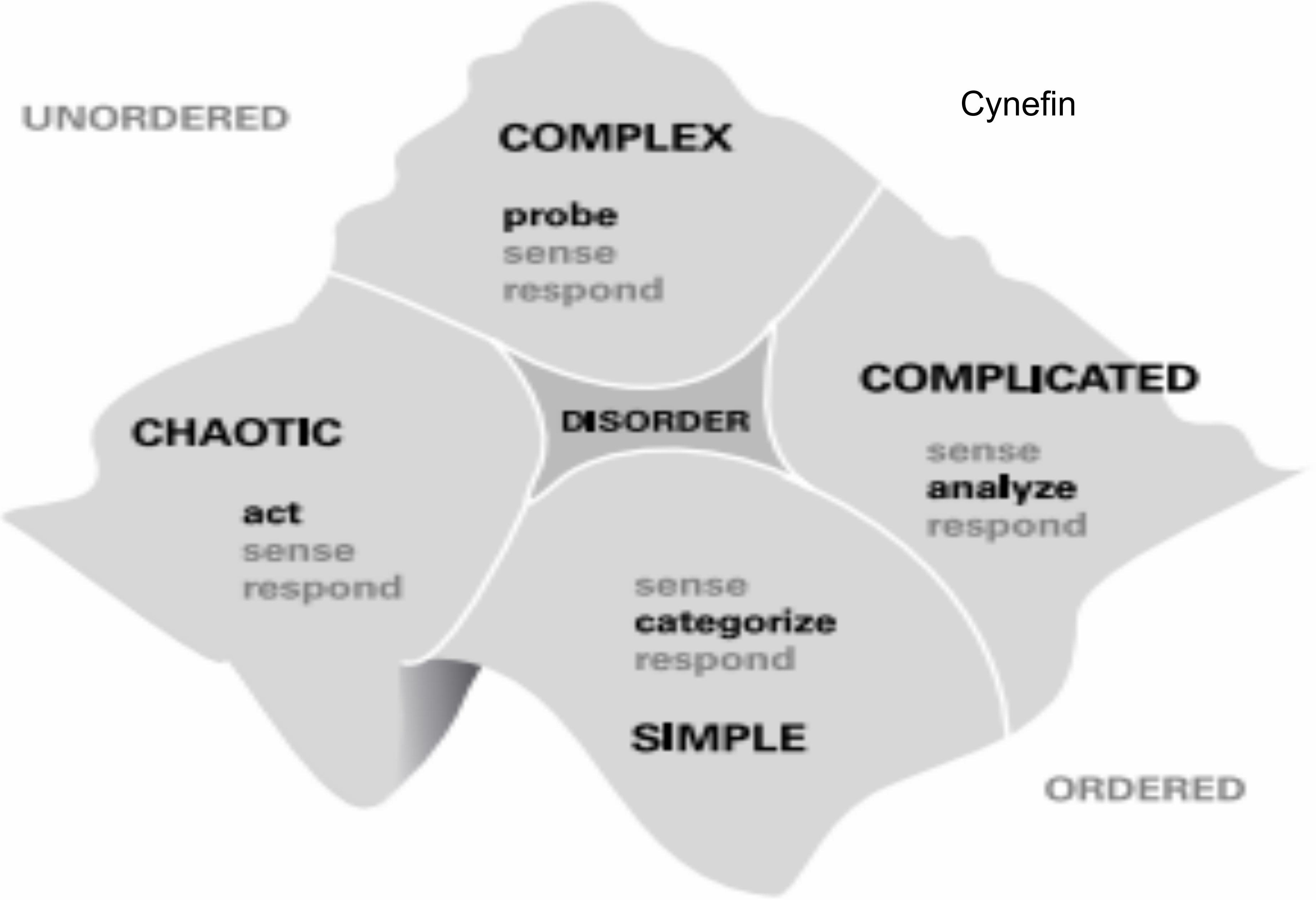
A3

PDCA - a basic element of the Total Quality Management movement.

The power of A3 reports and the thinking behind them requires a good grasp of the Plan-Do-Check-Act (PDCA)

The PDCA cycle simply follows the steps of the scientific method:
Plan is developing a hypothesis and experimental design;
Do is conducting the experiment;
Check is collecting measurements; and
Act is interpreting the results and taking appropriate action.

Cynefin


UNORDERED
COMPLEX
probe
sense
respond
COMPLICATED
sense
analyze
respond
CHAOTIC
act
sense
respond
DISORDER
sense
categorize
respond
SIMPLE
ORDERED

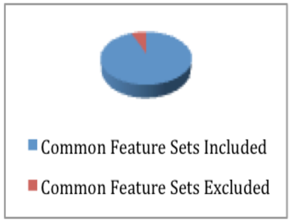
Background Plan

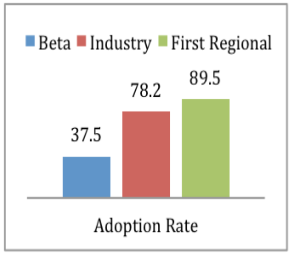
Mobile banking app paradox:

1. Lack of basic app directly attributed to Bank losing $100,000 per day (loss of existing customers + lost opportunity)
2. Pressure to develop and deploy app release is high
3. Limited beta release experiencing low adoption
4. Final feature set for full release intended to serve as a market differentiator

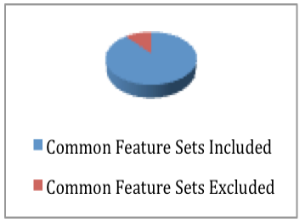
Current Condition Plan

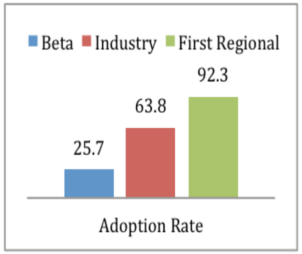
Non-Commercial Users





Commercial Users





Goal / Target Condition Plan

1. Identify factors driving low adoption rates.

What We Know Plan

2. Are there significant differences in functionalitycompared with other apps? NO – identical functionality independently confirmed.
3. Is the demographic of beta group skewed such that it represents a segment of the population not interested in this type of technology? NO – same group has readily adopted other mobile technologies at significantly higher rates.
4. Are there significant performance or quality issues with the app? NO – very positive feedback from existing users who have adopted. Performance and quality well within acceptable tolerances.
5. Is there a lack of awareness of the app among beta group? NO – Marketing division engaged in a structured campaign (similar to previous campaigns) that began several months before planned release date.
6. Have non-adopting customers ever logged on and used the app? YES – Most downloaded and logged in to the app when first offered.

Owner: Sameer

Mentor: Sanjay

Date: March 21, 2013

Countermeasures (Experiments) Do

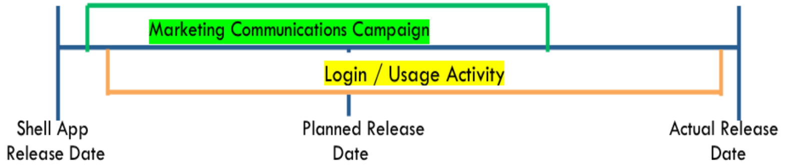
1. Analyze account activity Logs –.
    1. Patterns around initial usage (dates, frequency, etc.)
    2. Patterns around usage fall-off (timing, specific functions / requests, etc.)
2. Use push notification functionality within app to survey users of both active and inactive accounts.
    1. Patterns around banking behaviors
    2. Patterns around usage

Confirmation (Results) Check

1. Account activity logs
    1. Pattern of inactive accounts reflecting successive log ins after planned release date and before actual release.
    2. Pattern of inactive accounts reflecting successive API calls via other apps after actual release date.
2. Push notification surveys
    1. No discernable patterns around banking behavior based on responses provided by active and inactive accounts.
    2. Pattern of increased mobile app usage by 75% of inactive users that responded to push notifications.

Hypothetical Cause andEffect: 10 week delay between planned release date and actual release date had an adverse impact on adoption rates. Customers expectations were not met and they sought out third-party applications that provided some desired functionality.

Timeline:



Follow-up (Actions) Act

1. Closer coordination and communications with marketing communications division. Communications and outreach campaigns were based solely on the planned release date. 10 week delay between planned and actual release resulted in consumer confusion (and expectations not being met).
2. Improve process of making partial releases (if possible) by planned dates.

System 1 (Sam's Team)

6 Weeks

Iteration Backlog	Analysis		Develop		Test		Deploy	
Ready	Doing	Done	Doing	Done	Doing	Done	Ready	Done
🟨		🟨	🟨		🟨		🟪	
🟪			🟪					
🟨								
	DoD		DoD		DoD		DoD	

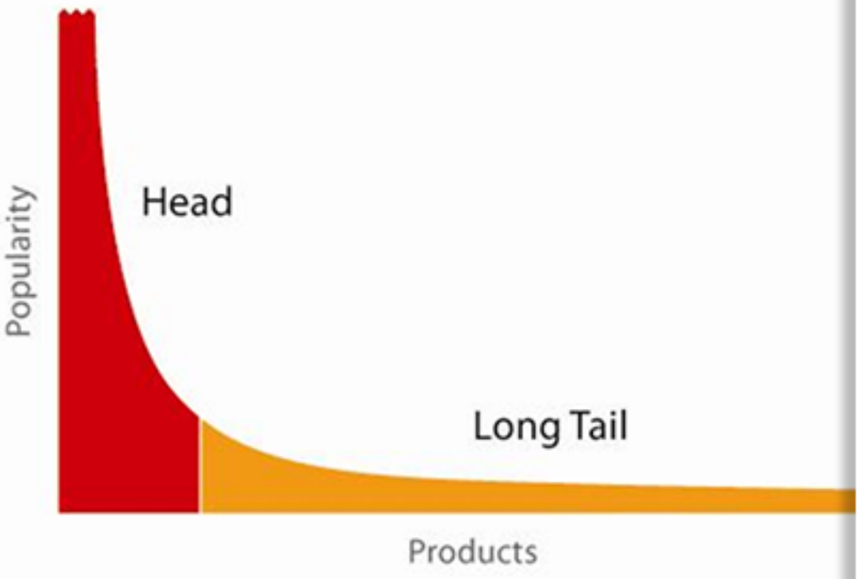
🟨 - Standard

🟪 - Bug F

- Visualization aided sharing of learnings with others- Collaboration gains
- Adopted Buffer Lanes
- Not handoffs but discovery of information
- Lean Coffee

Learning Evolutions

Mastery Level 🧍 🧍
A3 Thinking 🧍
Cynefin
Agile Budgeting
Long-Tail



Effect of "Long Tail" on Business alignment of the system

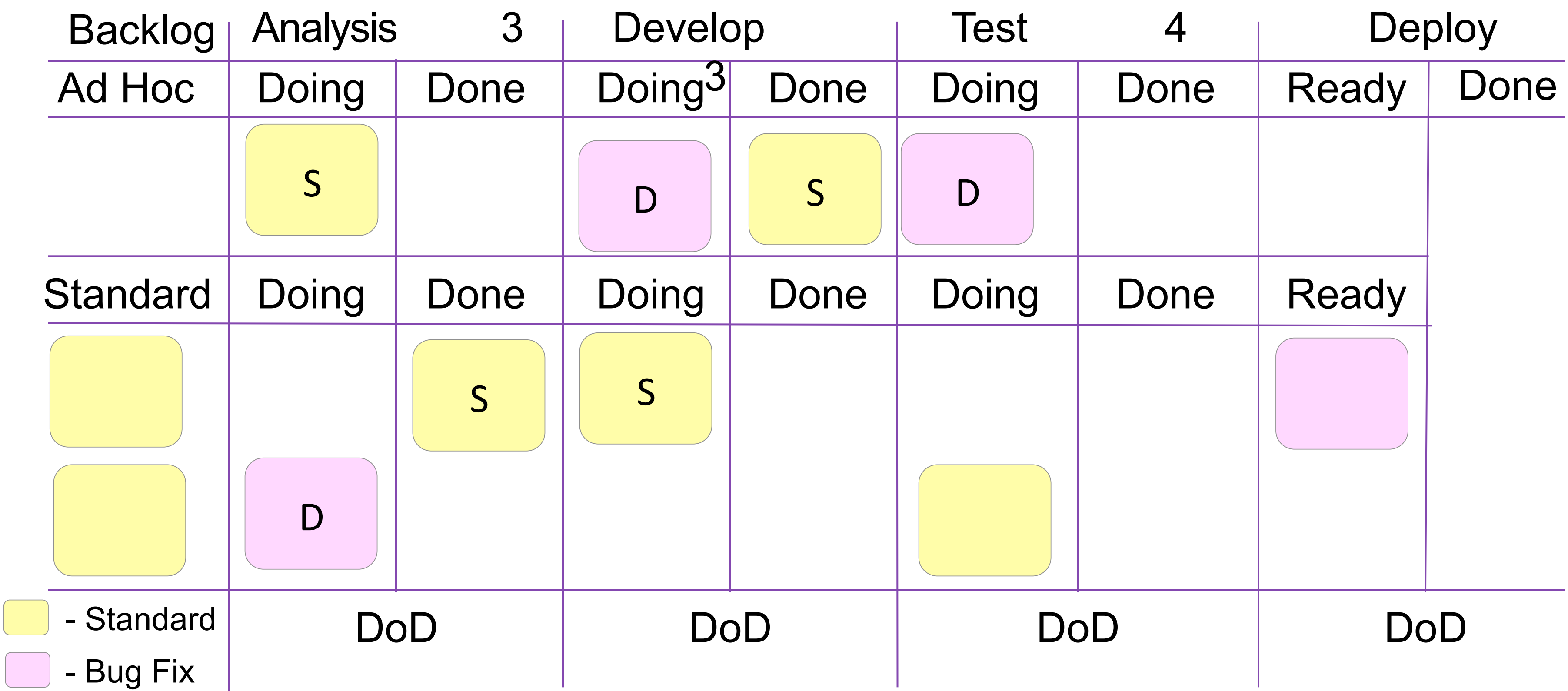
- Changes in Product Owner Options and Priorities – Niche features, Potential Value, Risk of
- Change Budgeting?

Differentiators, Spoilers, Cost Reducers, Table Stakes

Innovative, Major Growth Market, Cash Cow

System 1 (Sam's Team)

4 Months



Backlog	Analysis	3	Develop		Test	4	Deploy	
Ad Hoc	Doing	Done	Doing 3	Done	Doing	Done	Ready	Done
	S		D	S	D			
Standard	Doing	Done	Doing	Done	Doing	Done	Ready	
🟨 🟨	D	S	S		🟨		🟪	
	DoD		DoD		DoD		DoD	

🟨 - Standard

🟪 - Bug Fix

- Adopted Classes of Service & WIP Limits based on experience shared by Bala's team.
- Collaborated with Bala's team to apply A3 Thinking to problem of business needing to release features to the market as soon as possible.
- Identified Continuous Flow & Modified Deployment mode as solution (and successfully negotiated with the product owner and other dependencies).

Learning Evolutions

Mastery Level 🧍 🧍 🧍

A3 Thinking 🧍 🧍

Cynefin

Agile Budgeting

Long-Tail 🧍 🧍

System 4 (Sri’s Team)

8 Weeks

Iteration Backlog	Analysis	Develop	Test	Deploy	Done
🟪 D 🟨 S	🟨 C	🟨	🟨	🟪	
🟨	🟪	🟨	🟪		
		🟨			
	DoD	DoD	DoD		

Differentiators, Spoilers, Cost Reducers, Table Stakes

🟨 - Standard

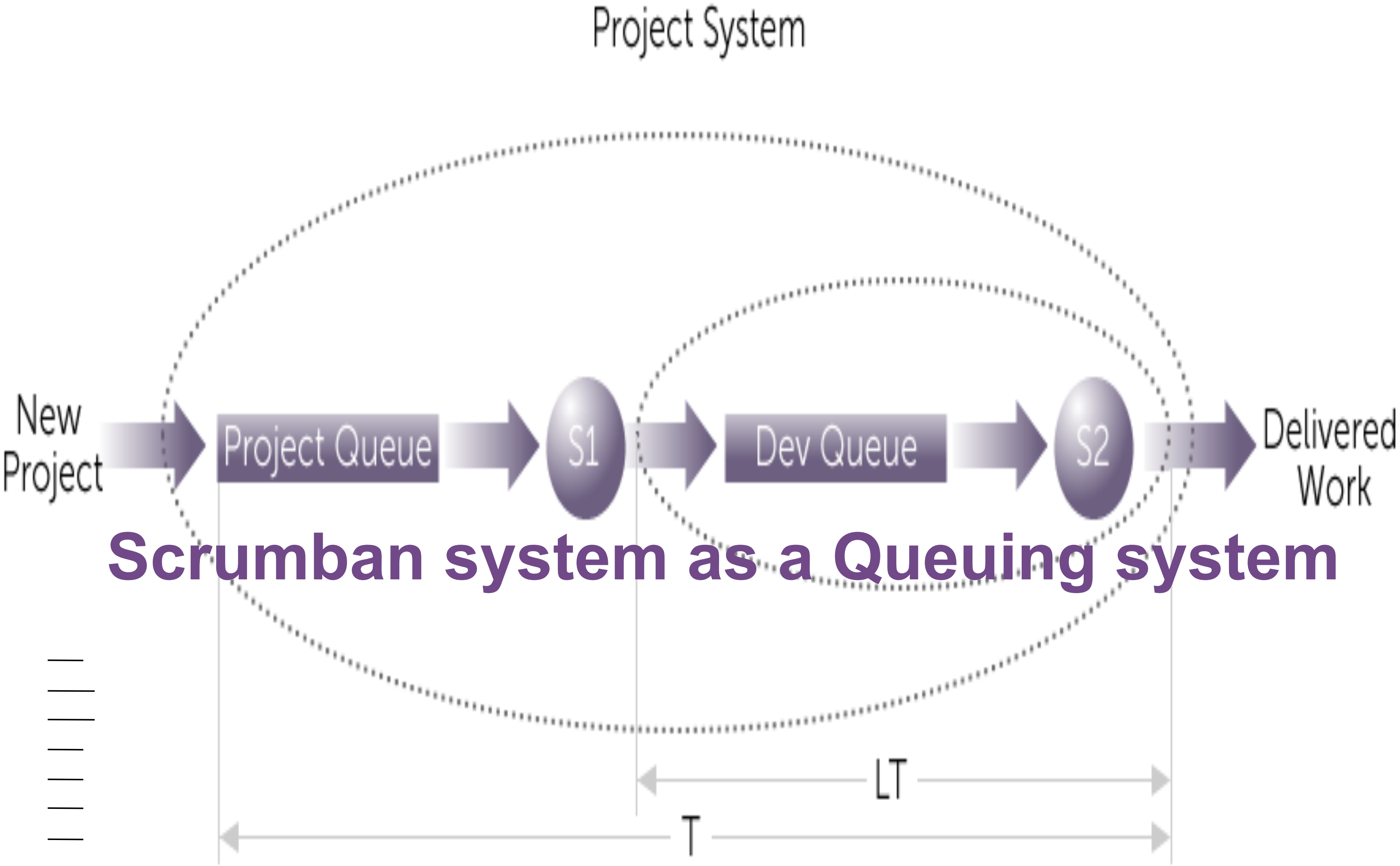
🟪 - Bug Fix

- Still practicing basic Scrum and using its visualization as a task board.
- Applying Team 1’s Long-Tail to prioritize backlog items because it was mandated by team leader.

Learning Evolutions

Mastery Level 🧍
A3 Thinking
Cynefin
Agile Budgeting
Long-Tail 🧍

Scrumban system as a Queuing system



Applying Little’s law

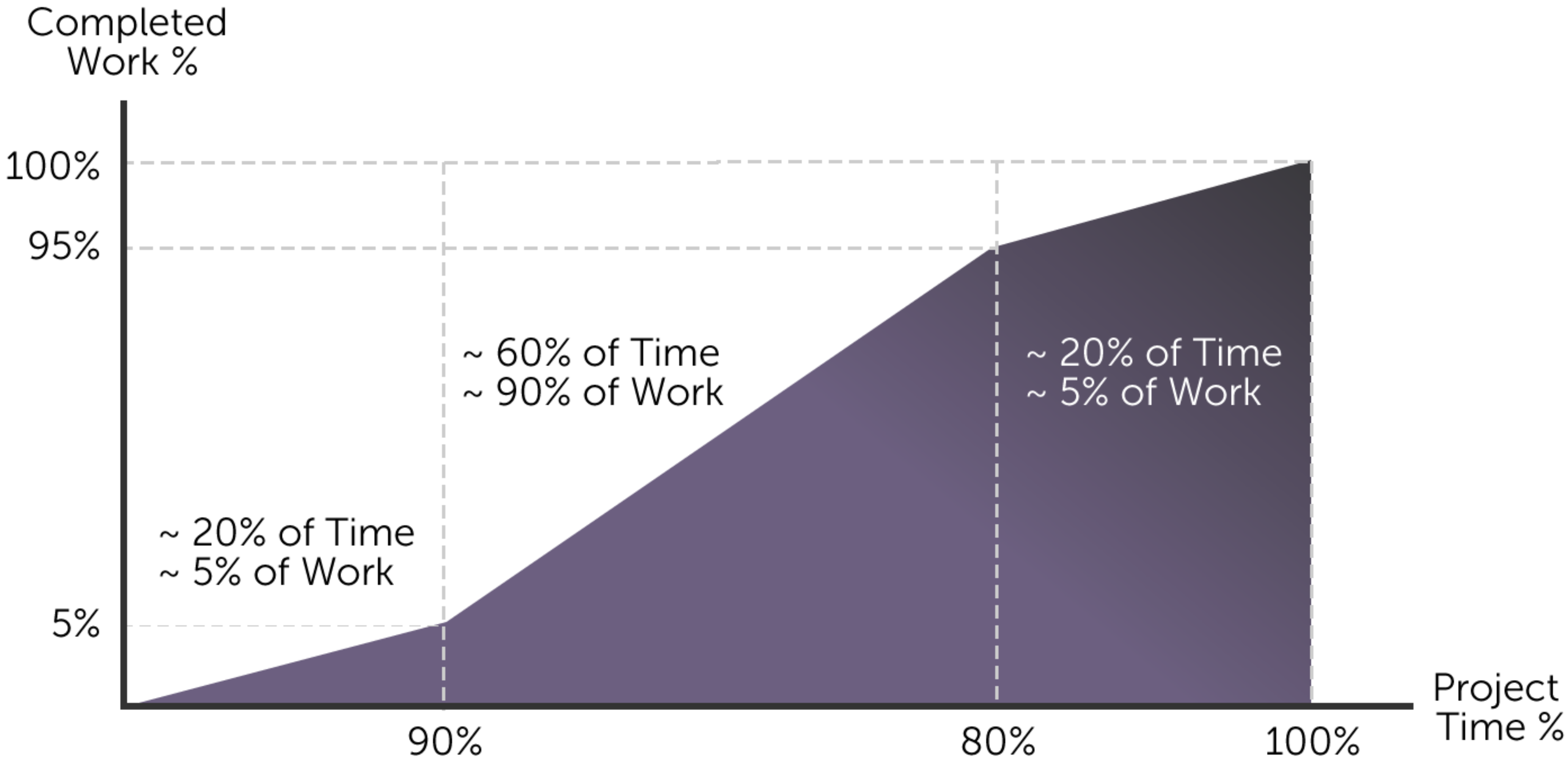
Little’s Law holds in case of a project:

Departure rate approximately equals arrival rate (λ).

No work items that get lost or never depart from the system.

It can be applied at an iteration level of a project or continuous flow.

Z Curve Breakdown



$(M-m)\int(\cos\varphi,$
$=(M-m)(-\sin\varphi,$
$+(M-m)(-\sin\varphi,$
$h_{4,1}(\varphi)=\pi/2+(1+d)(\varphi-\pi/2)-t$
$h_{3,4}(\varphi)=3\pi/2+(1-d)(\varphi-3\pi/2)+t$

Basic Sprint/Release Planning with Little's law

(Use with much care - Use qualitatively to influence/learn not make commitments)

We can calculate WIP to deliver for a particular lead time using the formula:

$$\overline{WIP}_t = \overline{LT}_t \frac{N}{T}$$

Managing uncertainty when planning a project

Uncertainty is reduced by aggregation. When calculating Project lead time we need to account for the uncertainty in terms of:

- Account for the first and third legs of the Z-curve
- Account for Dark Matter
- Account for Failure load

Source: Dimitar Bakardzhiev

On managing uncertainty

Buffer Management

Size Buffer

.

Scope Buffer

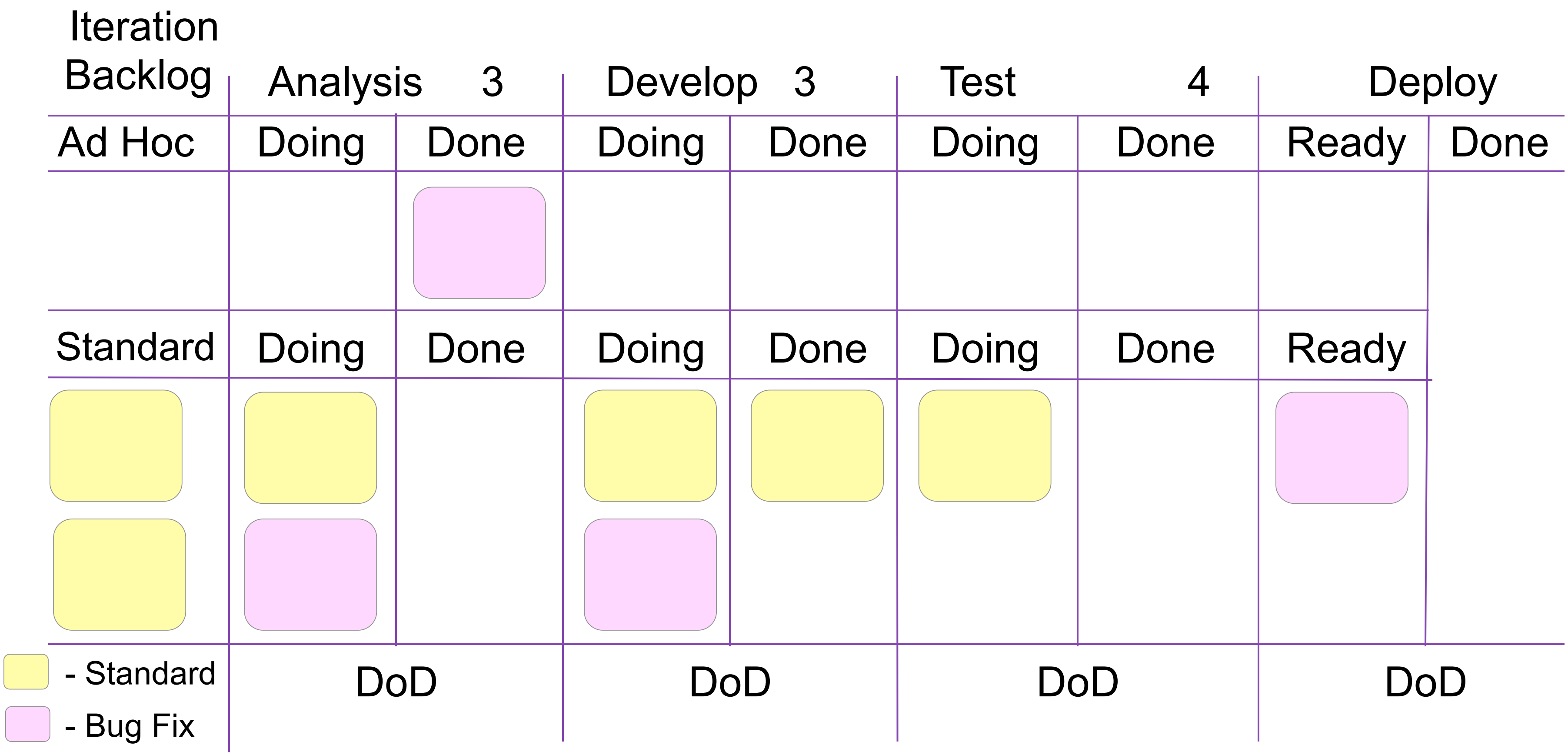
Time Buffer

Skills Buffer

Focus on median lead time

System 2 (Bella's Team)

8 Weeks



Iteration Backlog	Analysis 3		Develop 3		Test 4		Deploy	
Ad Hoc	Doing	Done	Doing	Done	Doing	Done	Ready	Done
		🟪						
Standard	Doing	Done	Doing	Done	Doing	Done	Ready	
🟨	🟨		🟨	🟨	🟨		🟪	
🟨	🟪		🟪					
	DoD		DoD		DoD		DoD	

🟨 - Standard

🟪 - Bug Fix

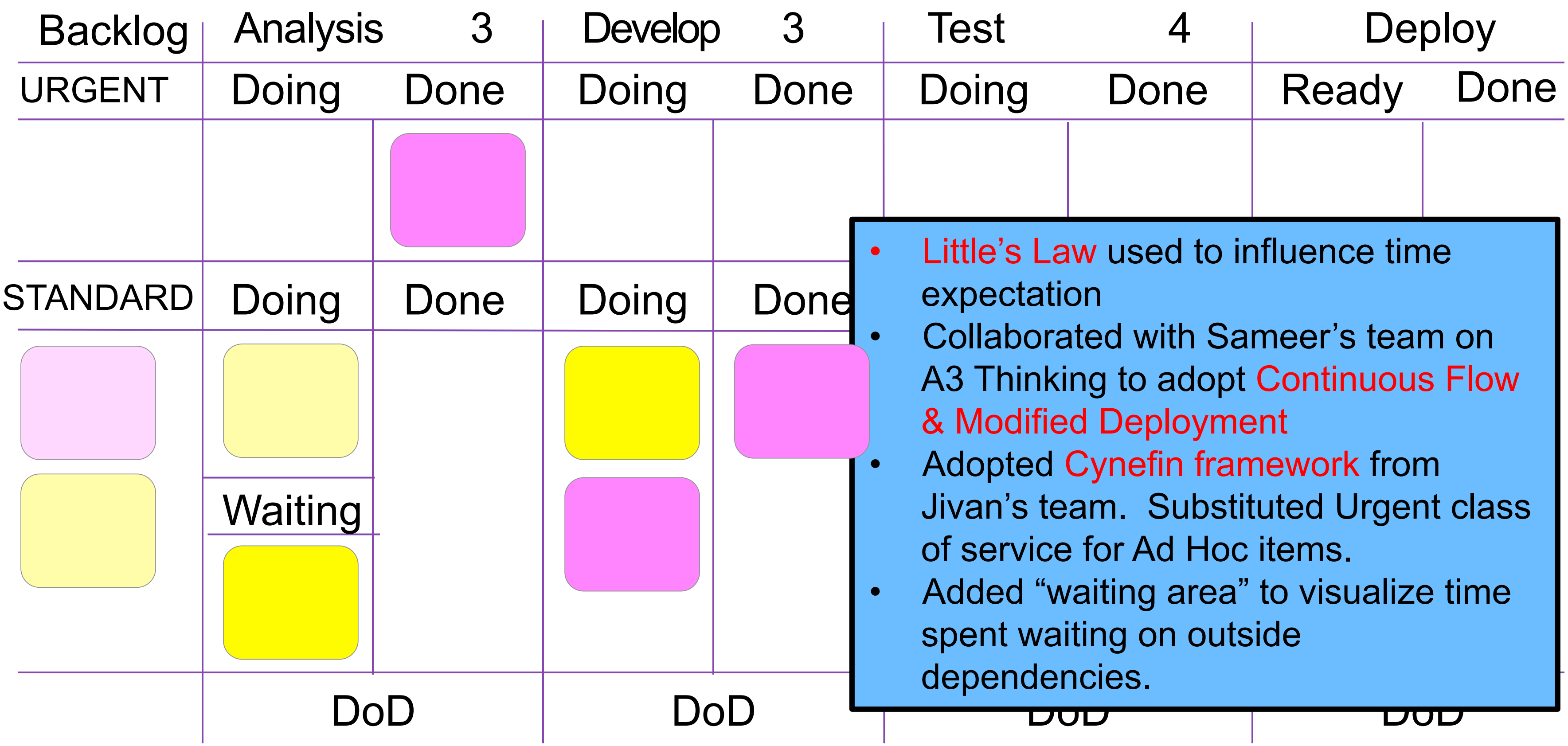
- Sharing its learnings with others, & applying Team 1's Long-Tail to prioritize backlog items.
- Adopted Buffer Lanes. WIP Limits & Classes of Service to differentiate ad hoc work requests.
- Unique in its efforts to plan with higher confidence Little's Law.

Learning Evolutions

Mastery Level 🧍 🧍
A3 Thinking
Cynefin
Agile Budgeting
Long-Tail 🧍

System 2 (Bella's Team)

4 Months



- Little's Law used to influence time expectation
- Collaborated with Sameer's team on A3 Thinking to adopt Continuous Flow & Modified Deployment
- Adopted Cynefin framework from Jivan's team. Substituted Urgent class of service for Ad Hoc items.
- Added "waiting area" to visualize time spent waiting on outside dependencies.

	Bug	Std
Simple	S	🟩
Complicated	🟪	🟨
Complex	🟪	🟨
Chaotic	🟥	🟥

Learning Evolutions

Mastery Level	🧍 🧍 🧍
A3 Thinking	🧍
Cynefin	🧍
Agile Budgeting	
Long-Tail 🧍	

System 5 (Raj’s Team)

4 Months

Iteration Backlog	Analysis		Develop		Test		Deploy	
Ready	Doing	Done	Doing	Done	Doing	Done	Ready	Done
🟨 🟪 🟨		🟨	🟨 🟪		🟨		🟪	
	DoD		DoD		DoD		DoD	

🟨 - Standard

🟪 - Bug Fix

- Adopted Buffer Lanes

Learning Evolutions

Mastery Level 🧍
A3 Thinking
Cynefin
Agile Budgeting
Long-Tail 🧍

System 1 (Sam’s Team)

6 Months

Backlog	Analysis	3	Develop		Tes		
URGENT	Doing	Done	Doing 3	Done	Doi		
	S		D	S	D		
STANDARD	Doing	Done	Doing	Done	Doing	Done	Ready
D	D	S	S	S	S		
FIXED/ INT	Doing	Done	Doing	Done	Doing	Done	Ready
	D						

- Adopted application of Cynefin framework & Classes of Service from Jivan’s & Bala’s teams.
- Adopts planning & forecasting with Little’s Law including Buffering.

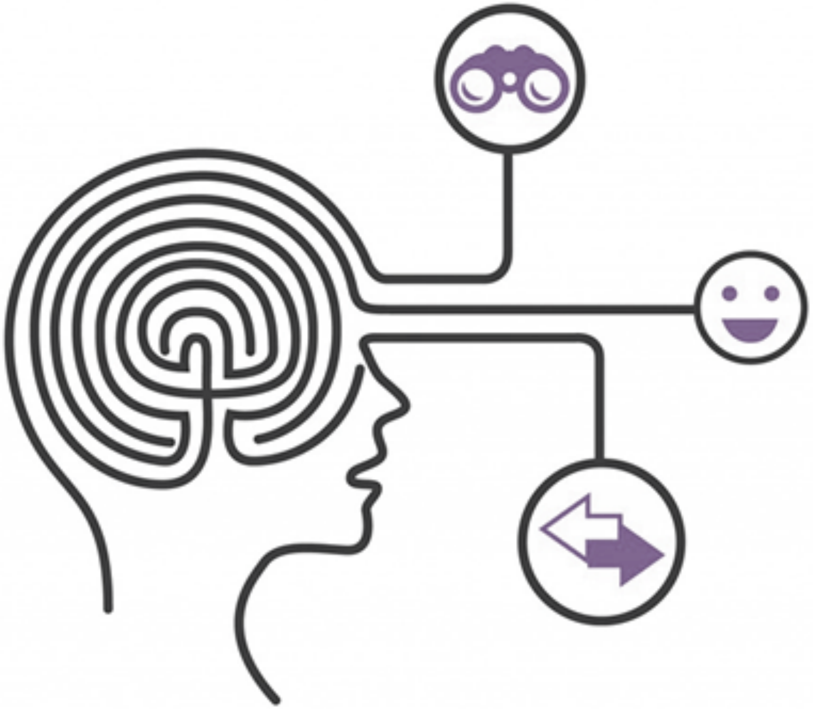
	Std
Simple	🟩
Complicated	▢
Complex	🟨
Chaotic	🟥

Learning Evolutions

Mastery Level	🧍 🧍 🧍
A3 Thinking	🧍 🧍
Cynefin	🧍
Agile Budgeting	
Long-Tail 🧍 🧍 🧍	

Design Thinking

Design thinking has come to be defined as combining empathy for the context of a problem, creativity in the generation of insights and solutions, and rationality in analyzing and fitting various solutions to the problem context.



Design Thinking

Consider

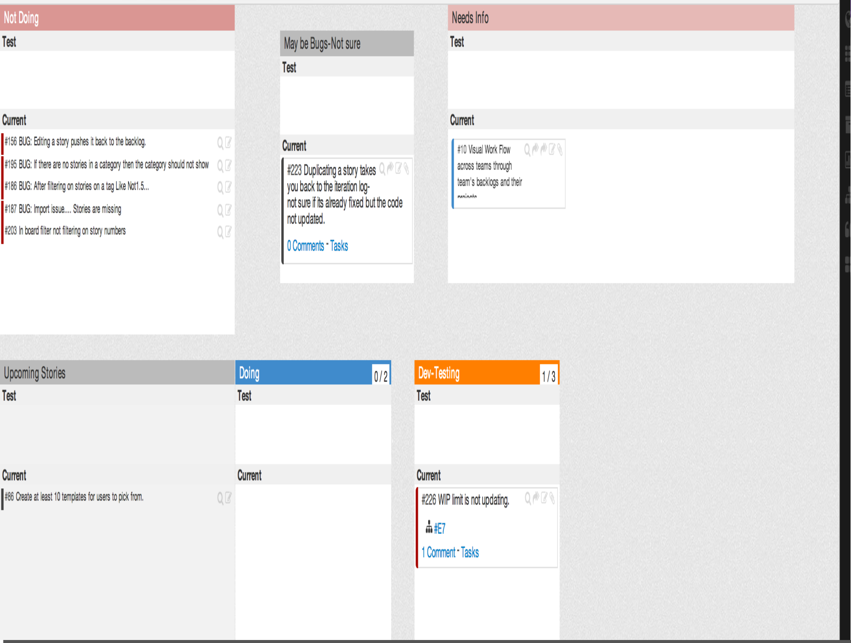
- Board Design
- Policy
- Distribute teams
- Electronic Tools

Design Thinking is "matching people’s needs with what is technologically feasible and viable as a business strategy"

Design Thinking

‘5 U’ Boards

- Uncluttered
- Up to Date at **all** times
- Universal
- Uniform
- Useful



Not Doing

Test

Current

#156 BUG: Editing a story pushes it back to the backlog.

#195 BUG: If there are no stories in a category then the category should not show

#186 BUG: After filtering on stories on a tag Like Not1.5...

#187 BUG: Import issue.... Stories are missing

#203 In board filter not filtering on story numbers

May be Bugs-Not sure

Test

Current

#223 Duplicating a story takes you back to the iteration log- not sure if its already fixed but the code not updated.

0 Comments - Tasks

Needs Info

Test

Current

#10 Visual Work Flow across teams through team's backlogs and their projects

Upcoming Stories

Test

Current

#86 Create at least 10 templates for users to pick from.

Doing 0 / 2

Test

Current

Dev-Testing 1 / 3

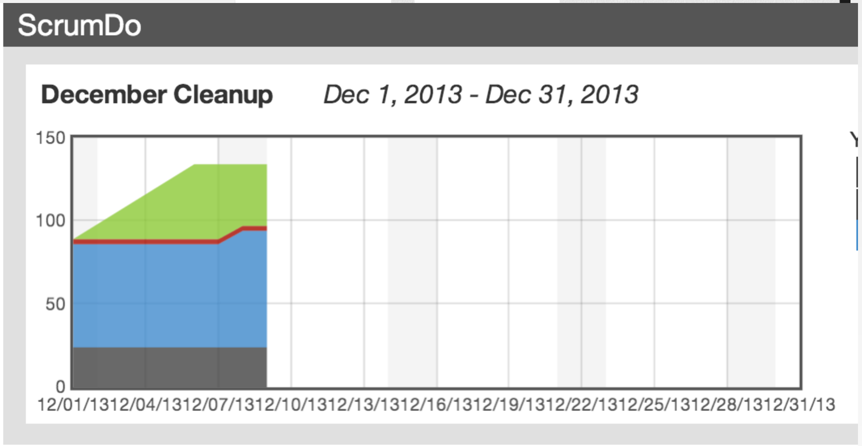
Test

Current

#226 WIP limit is not updating.

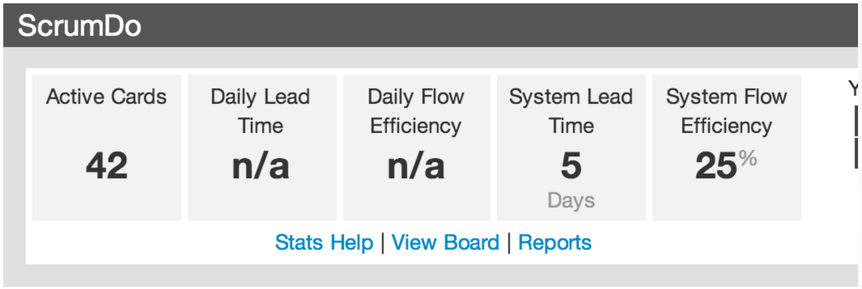
#E7

1 Comment - Tasks



ScrumDo

December Cleanup *Dec 1, 2013 - Dec 31, 2013*



ScrumDo

Active Cards	Daily Lead Time	Daily Flow Efficiency	System Lead Time	System Flow Efficiency
42	n/a	n/a	5 Days	25%

Stats Help | View Board | Reports

Options Theory

> “Never commit early unless you know why”
>
> *- Olav Maassen*

Options Theory

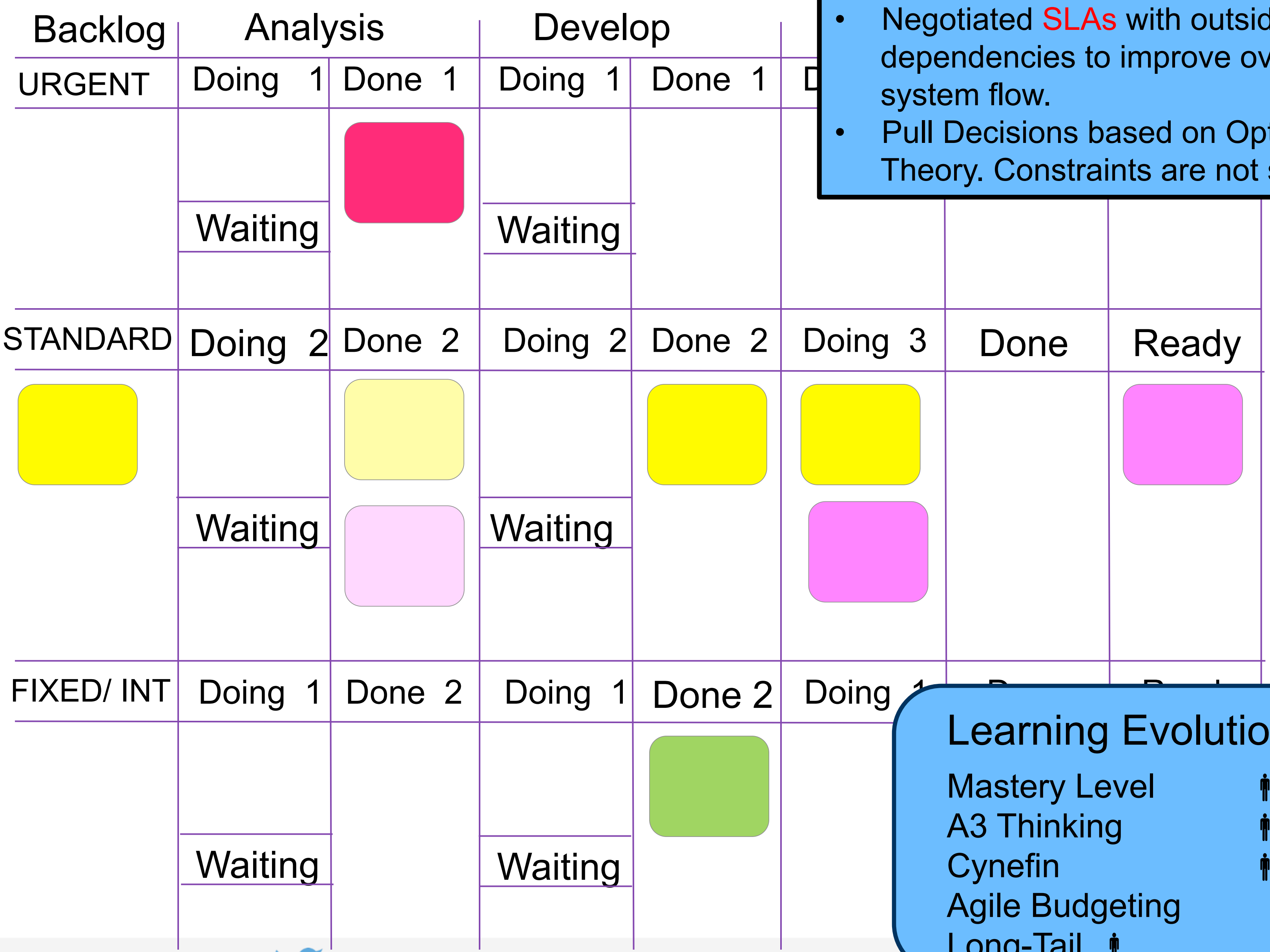
- Set your own constraints.
- Review your options.
- Collect information (and find more options).
- When an option is about to expire review your constraints to see if they are still valid.
- Choose the best option based on your current information at hand.

Options Theory

Consider

- “Iteration commitment”
- “Cost of Delay” of releasing a product
- “Last Responsible Moment”

System 2 (Bella's Team)



Backlog

Analysis

Develop

URGENT

Doing 1

Done 1

Doing 1

Done 1

D

Waiting

Waiting

STANDARD

Doing 2

Done 2

Doing 2

Done 2

Doing 3

Done

Ready

Waiting

Waiting

FIXED/ INT

Doing 1

Done 2

Doing 1

Done 2

Doing 1

D

Read

Waiting

Waiting

- Expanded dependency waiting areas to all effected phases.
- Negotiated SLAs with outside dependencies to improve overall system flow.
- Pull Decisions based on Options Theory. Constraints are not static.

Learning Evolutions

Mastery Level	🧍 🧍 🧍
A3 Thinking	🧍 🧍
Cynefin	🧍
Agile Budgeting	
Long-Tail 🧍	

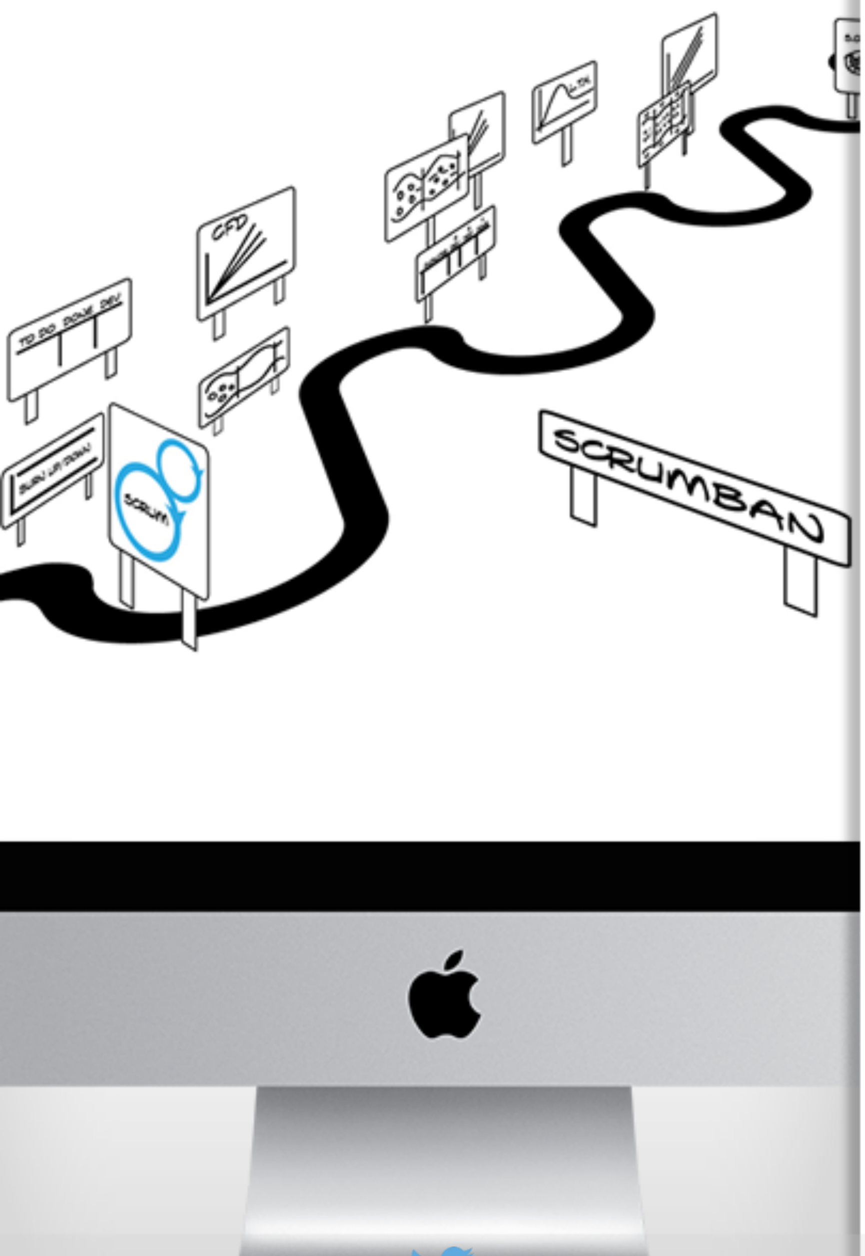
Scrumban

False Hybrids and common 'macro' evolutions

"Scrum in Kanban"

"Kanban in Scrum"

Scrumban is not a process process framework or a process destination!



Scrumban Definition

Scrumban is the Kanban method applied to Scrum as the starting and underlying process framework.

Scrumban is a **pragmatic** and **entrepreneurial** application of The Kanban Method to Scrum, using scientific **management theories** often resulting in a **continuing evolution** of Scrum based systems into business valued, service oriented, Agile, Lean and adaptive systems at a **pace moderated** by the thinking systems.

Do More

ScrumbanDo

ScrumDo’s new product release

Scrumbando.com

Codegenesys.com/events

Foundations training

Advance Practitioner Training

Scrumban Guarantee program

Contact

info@codegenesys.com

Like Us

facebook

https://www.facebook.com/scrumban

Section 4 of 4

Scrumban fitness

Measurements

Why measure ?

- Measuring the right things and giving it the appropriate weight
- Measuring with appropriate degree of certainty
- Making sense of the data- Discipline of collection and Analysis

Measurements

> Any observed statistical regularity will tend to collapse once pressure is placed upon it for control purposes.

-Goodhart

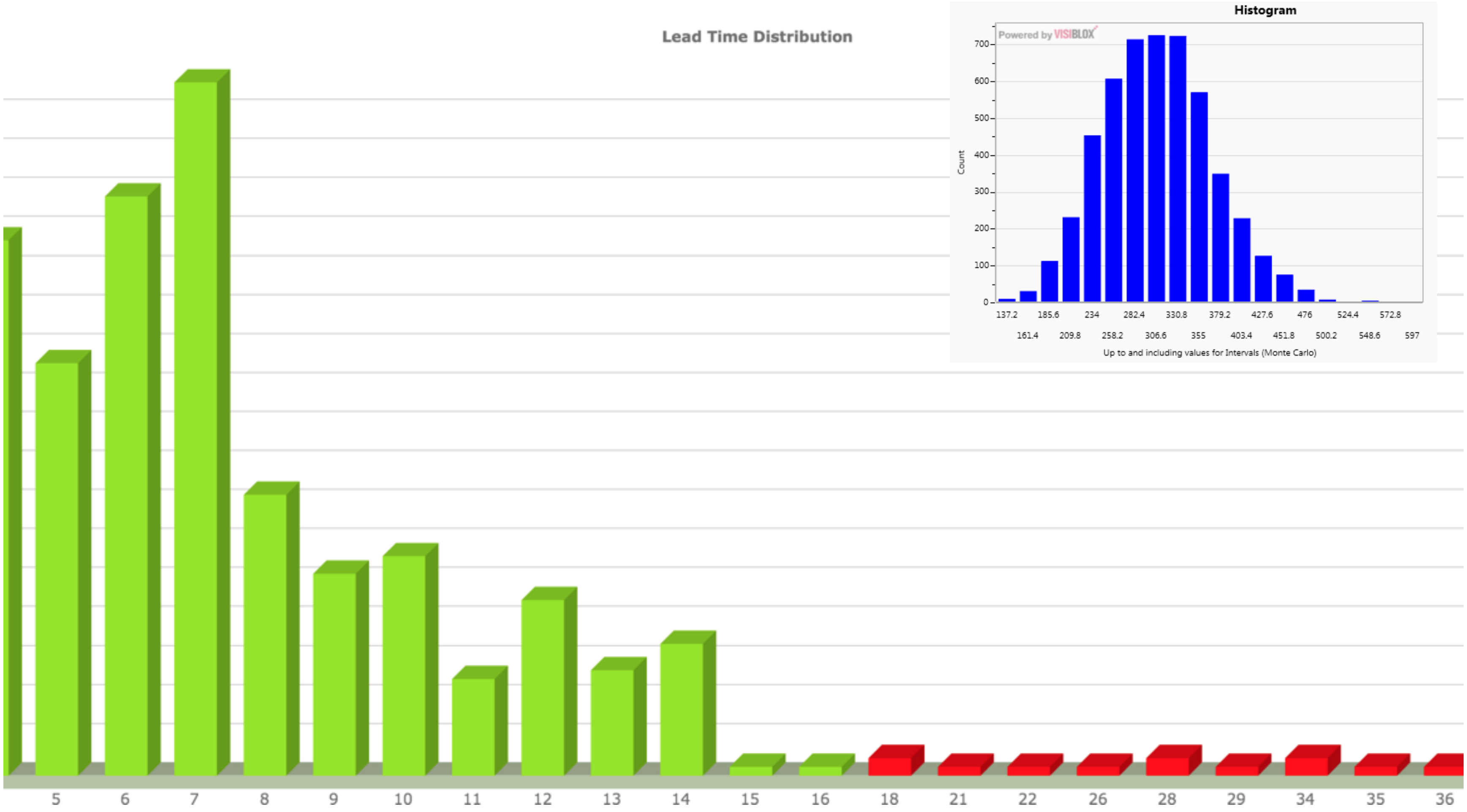
Measurement types

- Qualification vs Quantification
- Nominal vs Ordinal
- Deterministic vs Probabilistic
- Domain specific vs Domain agnostic

Scrumbando

- Lead Time
- Flow Efficiency
- Customer Lead Time
- Touch Time
- Takt Time
- LTH
- CFD
- Throughput
- Aging of WIP
- SLA Performance
- Due Date Performance
- Cause of Failure Load
- Blocked CFD
- Story Points CFD
- Points to Lead time
- More

Lead Time Historgram

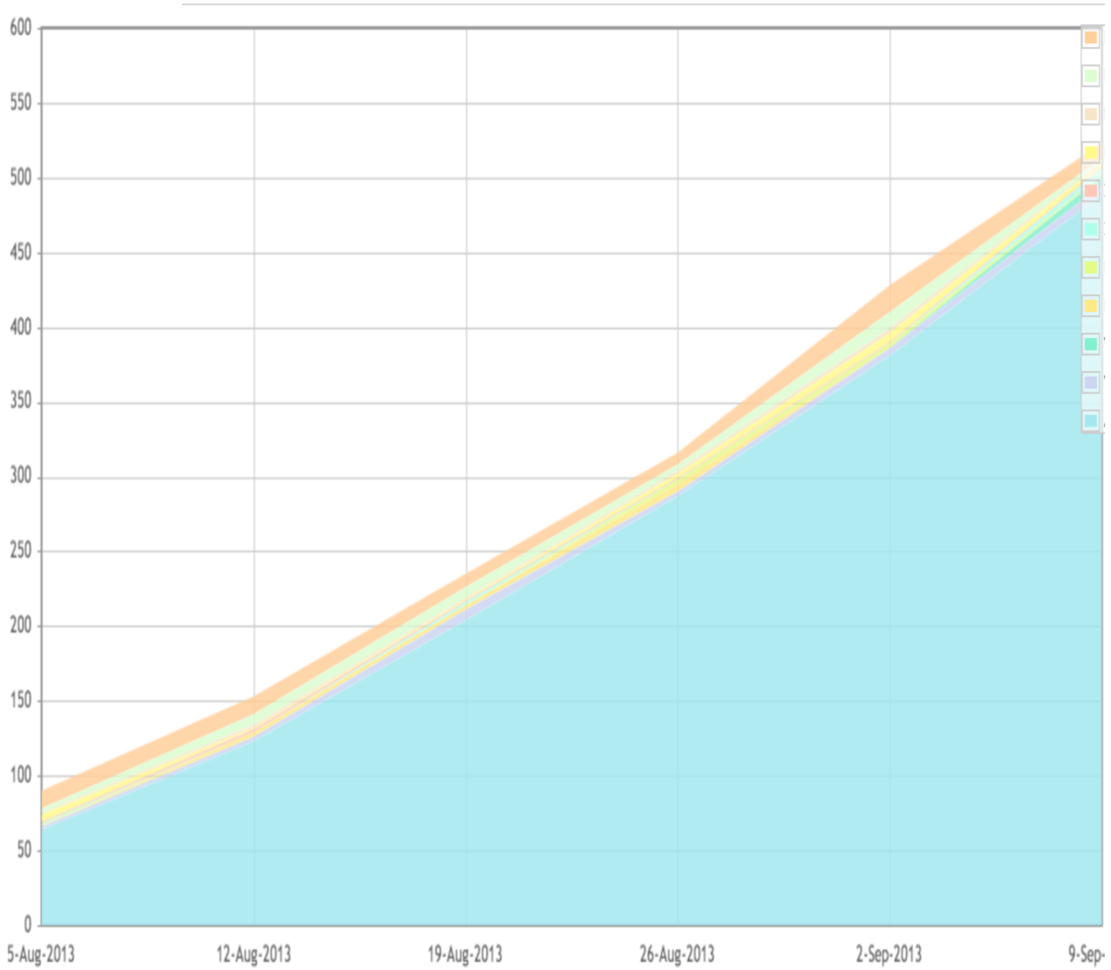


CFD

CFD shows the quantities of work in progress (WIP) at each stage in the system.

CFD is a visual representation of Little's Law.

Average Lead Time = Average Work in Progress / Avg Throughput



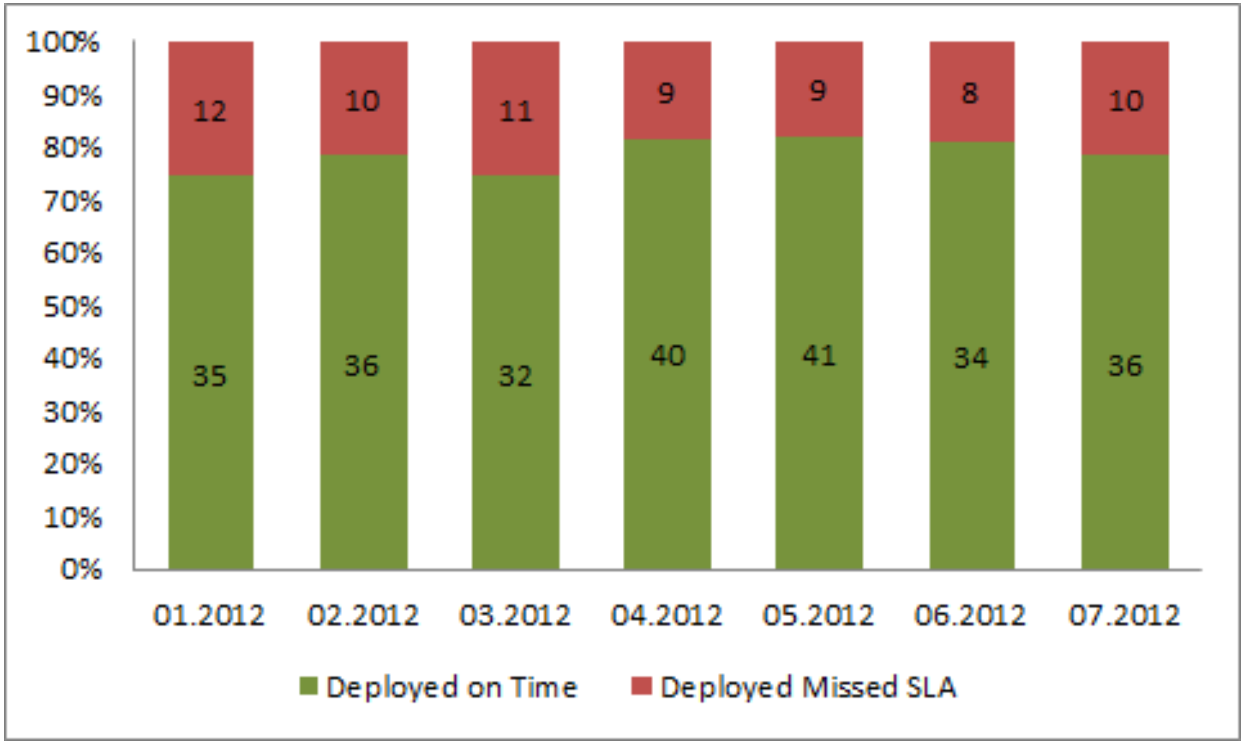
Flow Efficiency

- Calculated as touch time against lead time.
- It indicates how much room there is for improvement by eliminating waste without changing the engineering methods.

5-40%

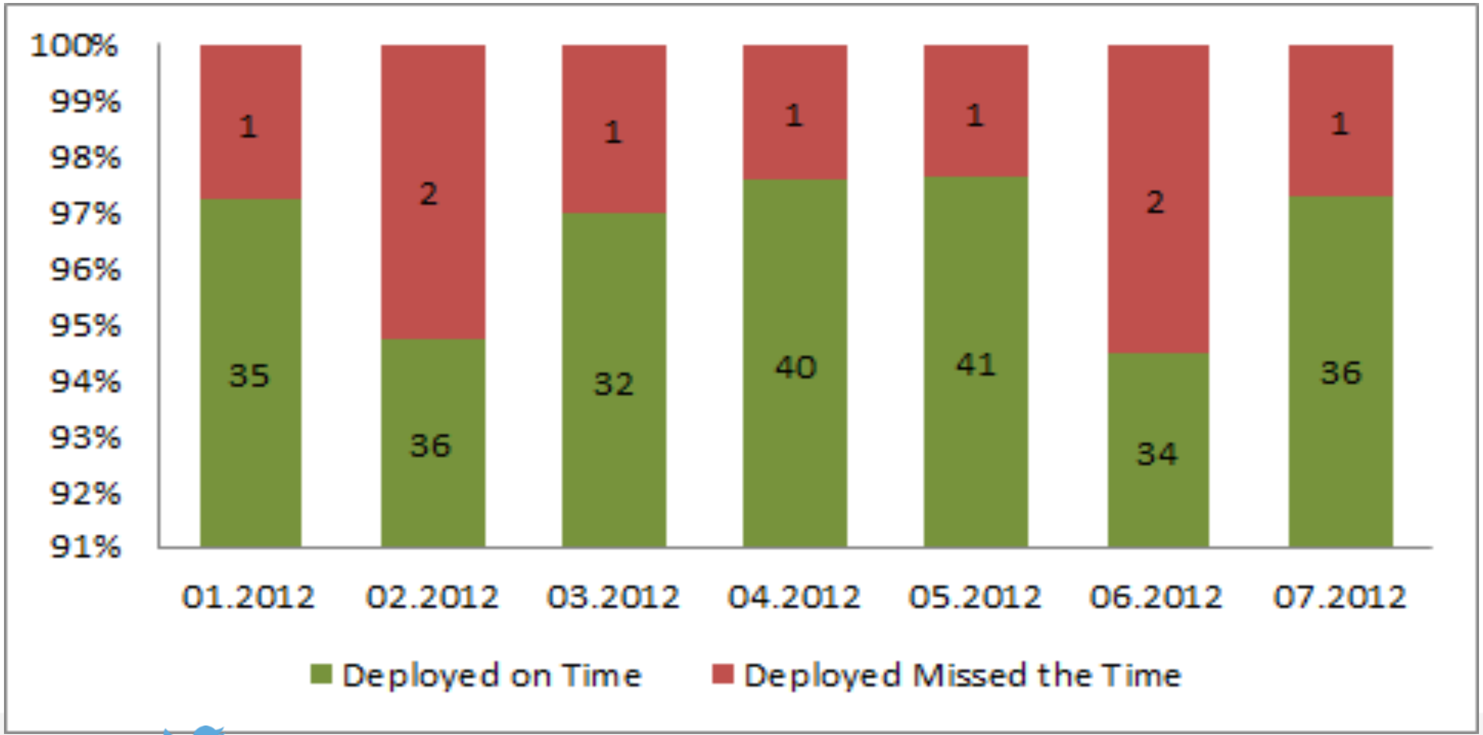
SLA Performance

- Classes of Service spread risk



Due Date Performance

- For Fixed Date class of service we should show the answer of the question “Was the work item delivered on time?”



Successful outcomes at Mammoth bank

Though not all groups ultimately applied Scrumban to their way of working, it was ultimately adopted by the vast majority of teams within the RAD Group, and had also begun to spread to other groups across the larger IT organization within a few short months. It was recognized as a a simple, easy-to-use, yet powerful framework for identifying and managing impediments to workflow.

Successful outcomes at Mammoth bank

Though the RAD Group did not make up for all of its lost time on the mobile banking app, it ultimately delivered the second and final release just 5 weeks past it’s original due date. As the first release had been 10 weeks past its due date, this was viewed as a major achievement.

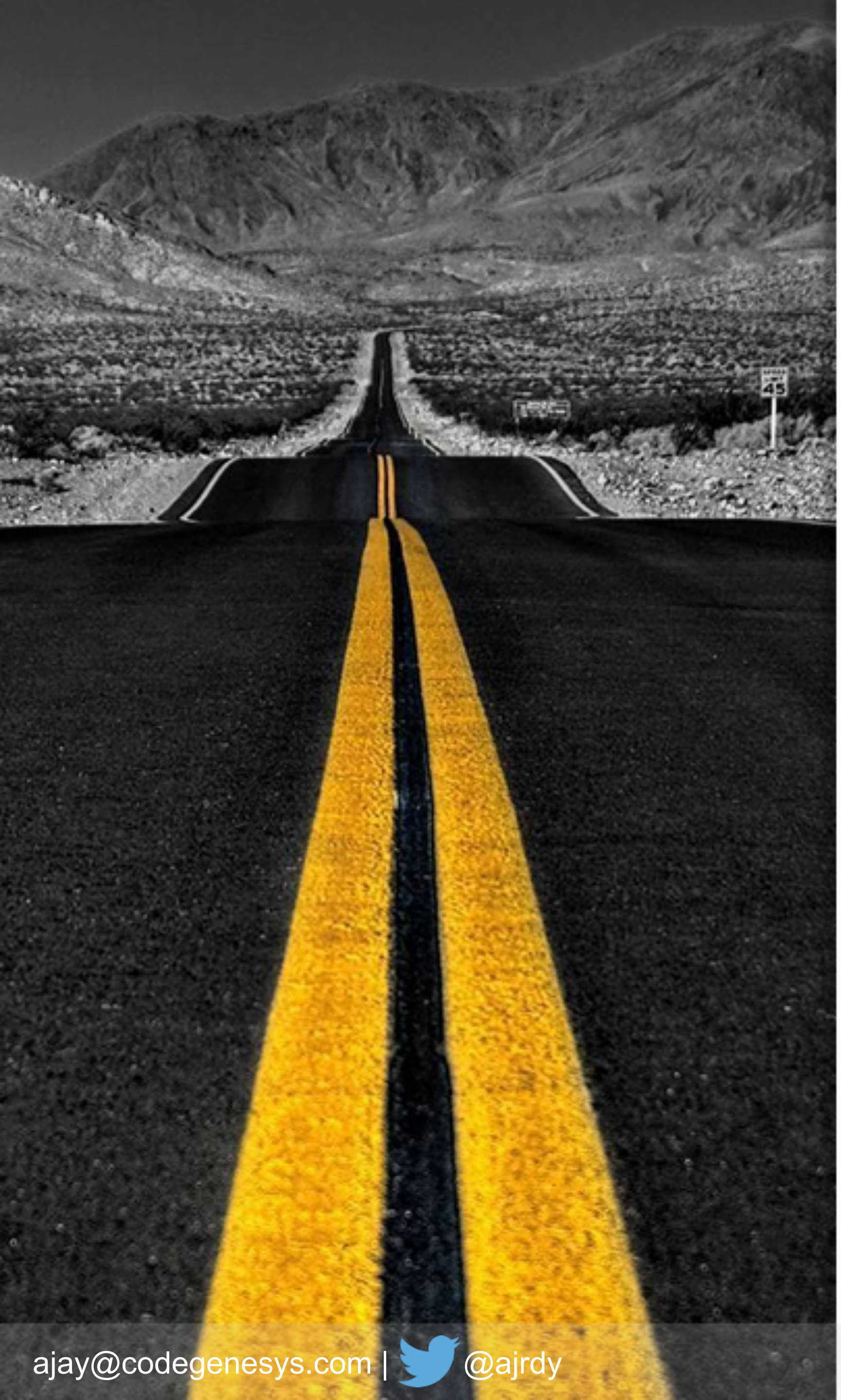
Monitoring Projects in flight

Planned Project lead time is the sum of the calculated project length and a project buffer

The two essential measurements of project performance are the percentage of the project completed and the amount of the project buffer consumed.

“ The most important things cannot be measured ”

-Deming, W. Edwards. Out of the Crisis.



More

Management

- How do roles, leadership work out in Scrumban setups

Maturing

- What are some other catalysts in maturing

Modeling

- Monte carlo simulations, Risk calibration

Scrumban.io

- Lots of contributions from many Scrum and Kanban leaders who really understand Scrumban

Like Us

facebook

https://www.facebook.com/scrumban

Do More

ScrumbanDo

ScrumDo’s new product release

Scrumbando.com

Codegenesys.com/events

Foundations training

Advance Practitioner Training

Scrumban Guarantee program

Contact
info@codegenesys.com