



Waterfall and Agile and Lean, Oh My!

September 2021

Michelle Johnson, PMP, CSM, Service & Platform Management Manager
Sharon Hayward, PMP, CSM, Lead Project Manager



UNIVERSITY OF
NOTRE DAME

History/Background



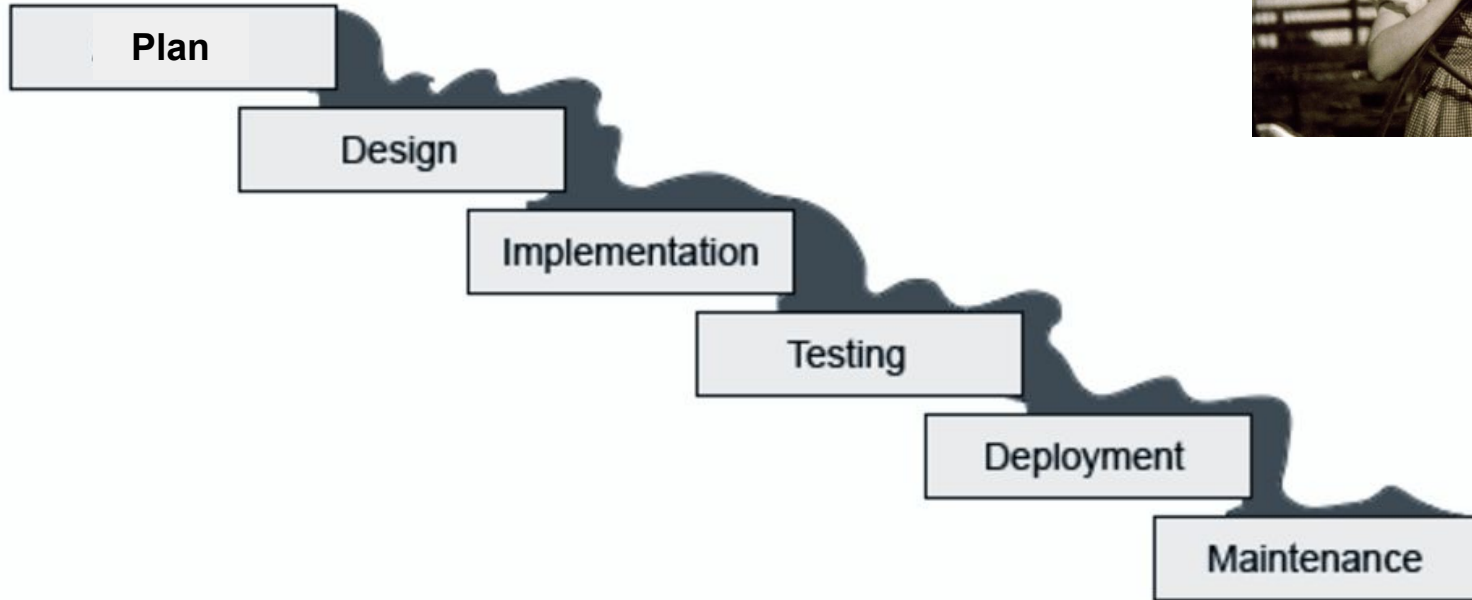
Purists



Absolute adherence to
traditional rules,
structures and...

PROJECT
METHODOLOGIES

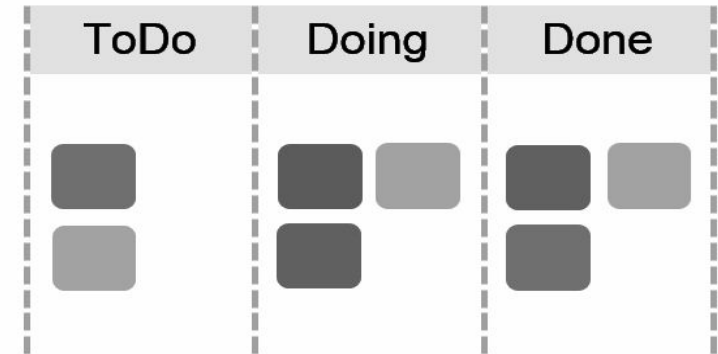
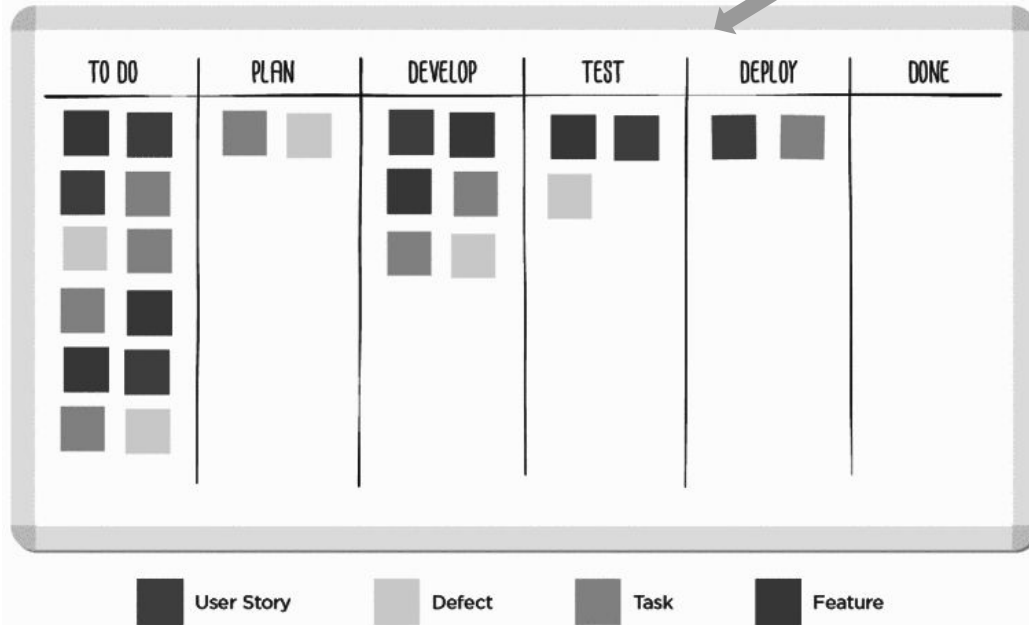
Waterfall



Maurya Rieder (March 3, 2017) Waterfall, Agile or Critical Chain - which methodology is right for you?.
Retrieved from: <https://www.linkedin.com/pulse/waterfall-agile-critical-chain-which-methodology-right-rieder>

Kanban

Lifecycle
Phases



LeanKit Inc (2017) What is a Kanban Board?. Retrieved from: <https://leankit.com/learn/kanban/kanban-board/>

Scrum



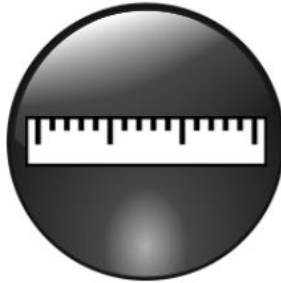
Jonathan Rasmusson (January, 2014) Agile in a Nutshell.
Retrieved from: <http://www.agilenutshell.com/scrum>

Lean DMAIC



DEFINE

Define the problem.



MEASURE

Quantify the problem.



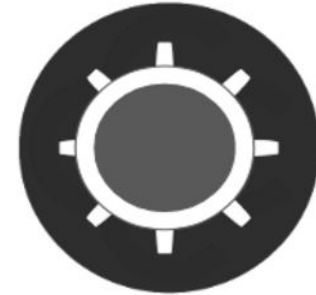
ANALYZE

Identify the cause of the problem.



IMPROVE

Identify and Implement the solution.



CONTROL

Maintain the solution.



Methodology Comparison

Framework	Planning	Execution	Operationalize
Waterfall	Requirements Document & Detailed Plan	Design, Develop, Test & Deliver	Closeout
Scrum	User Stories & Release Plan	Sprint Planning, Iterate, Demo & Retro	Closeout Sprint
Kanban	Minimum Marketable Feature & Release Plan	Flow Cards through Lifecycle	Final Support Phases
DMAIC (Greenbelt)	Define/Measure/Analyze/ Improve (Scoping)	Improve (Implement)	Control





From Kansas to Oz

Kanban

Without Work In Progress (WIP) limits



Scrum

With multiple product owners and epics vs stories



Waterfall

Without having full requirements or full sponsorship



DMAIC

Without a grand vision or discrete efficiency goals



Blending

Scrum & Kanban =

Scrum & Waterfall =

DMAIC & Kanban =



Scrum - Ban



Scrum - Fall



DMAI - Ban

Key Factors

1



2



3



4



5



6



Scrum-Fall



Users,
including
“Why”

- Customer - heavier involvement in Planning
- Team - works together to complete stories regardless of roles
- Project - Few unknowns and one big bang delivery

Framework	Planning	Execution	Operationalize
Waterfall	Requirements Document & Detailed Plan	Design, Develop, Test & Deliver	Closeout
Scrum	User Stories & Release Plan	Sprint Planning, Iterate, Demo & Retro	Closeout Sprint
Kanban	Minimum Marketable Feature & Release Plan	Flow Cards through Lifecycle	Final Support Phases
DMAIC	Define/Measure/Analyze/Improve (Scoping)	Improve (Implement)	Control

Scrum-Fall: Student Activities Online



Requirements:
SCRUM stories



Schedule: Sprints
transitioned to
Waterfall



Build: Iterations → Dated
tasks

Testing: Done with each
sprint; separate round at
end for Waterfall



Delivery: Throughout
project with final at end.



Stabilization: Scheduled
support activities



Water-Ban



Functional Specification



- Customer - not heavily involved
- Team - specialized, only used in certain stages
- Project - requirements are known upfront; scope prioritized by the team

Framework	Planning	Execution	Operationalize
Waterfall	Requirements Document & Detailed Plan	Design, Develop, Test & Deliver	Closeout
Scrum	User Stories & Release Plan	Sprint Planning, Iterate, Demo & Retro	OR Closeout Sprint
Kanban	Minimum Marketable Feature & Release Plan	Flow Cards through Lifecycle	Final Support Phases
DMAIC	Define/Measure/Analyze/Improve (Scoping)	Improve (Implement)	Control

Water-Ban: Going Google



Requirements:

Defined upfront



Schedule: Based on two delivery dates (early adopters & everyone else)



Build: Constant Flow using an Electronic Kanban Board

Testing: Three cycles of testing scheduled



Delivery: Two separate go-live dates



Stabilization: Included as discrete lifecycle stage



Decision Matrix - Reaching the Crossroads

Dorothy Gale:

Now which way do we go?

The Scarecrow:

Pardon me, this way is a very nice way.

It's pleasant down that way, too.

Of course, some people do go both ways.



Planning

- 1 Level of customer participation on the project team

Full	Often	Sometimes	Intermittent
	x	x	

- 2 Level of requirement/user story details needed prior to the start of execution

High	Medium High	Medium	Low
	x		

- 3 Project team methodology preference/experience

Waterfall	DMAIC	SCRUM	Kanban
x		x	x

- 4 Primary project driver

Process	Timeline	Unfamiliar	Functionality
		x	x

- 5 Ongoing measures required post project

Required	Recommended	Optional	Rarely/ N/A
			x

Planning Methodology

DMAIC	Waterfall	SCRUM	Kanban
0	2	3	3

Scrum
Kanban

Execution

6 Level of team oversight needed

High	Medium	Low
	x	

7 Ability to respond to changes to scope, timeline, and/or resources

Easy	Moderate	Difficult
	x	

8 Level of dependencies within the project

Many	Some	Few
	x	

9 Level of 3rd party vendor management (e.g. RFP)

High	Medium	Low or N/A
	x	

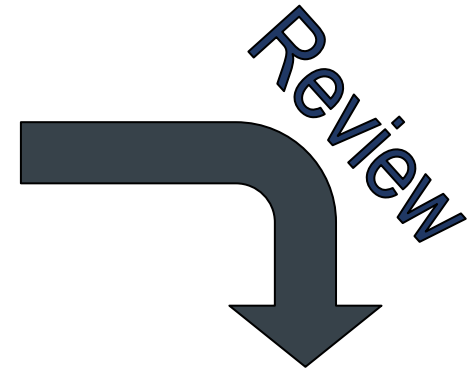
10 Delivery approach

Iterative	Constant Flow	Big Bang
x	x	

Execution Methodology		
Waterfall	SCRUM	Kanban
2	2	3

Kanban

Analyze the Results



Blend Matrix

Planning Methodology			
DMAIC	Waterfall	SCRUM	Kanban
0	2	3	3



Execution Methodology		
Waterfall	SCRUM	Kanban
2	2	3

Kanban -OR- Scrum-Ban

Name	When to use	Point of View	Requirements method	Schedule definition	Build	Testing	Delivery approach	Closing approach
DMAI-Ban <small>Blend of DMAIC and Kanban</small>	Customer concerned about delivery Process driven Team members are specialized, only used in certain stages Stories can be passed as needed without causing significant delivery effect	Organization efficiency	Requirements and development of the scope follows the defined DMAIC process. Define, Measure, Analyze, they are written in business language	Requirements are determined at a constant flow	Team defines the lifecycle stages Scheduling based on process, capacity and area of team resources Constant flow of work pulled by the team	Iterative testing lifecycle stage identified by the lifecycle stage Control measures exist and are monitored in "monitor" lifecycle stage Items requiring modification return to Develop/Build lifecycle stage	Control measures exist and are monitored in "monitor" lifecycle stage Control measures exist and are monitored in "monitor" lifecycle stage Control measures exist and are monitored in "monitor" lifecycle stage Control measures exist and are monitored in "monitor" lifecycle stage	Control measures exist and are monitored in "monitor" lifecycle stage Control measures exist and are monitored in "monitor" lifecycle stage Control measures exist and are monitored in "monitor" lifecycle stage Control measures exist and are monitored in "monitor" lifecycle stage
DMAI-Fall <small>Blend of DMAIC and Waterfall</small>	Intermittent customer involvement Process driven Can include intermittent vendor RFP May only have one big bang delivery	Organization efficiency	Requirements and development of the scope follows the defined DMAIC process. Define, Measure, Analyze, they are written in business language	Sequential task planning, considering dependencies, team resources and other standard waterfall planning -managed in traditional project management schedule or spreadsheet	Traditional waterfall Design/Build/Develop	Will have defined test plan to include different testing types Full and to end testing done prior to acceptance	Big bang delivery, which may have several smaller delivery points after the supporting functionality is accepted Control measures exist and are monitored	Stabilization, documentation, standard closing activities done at end of the project Control measures exist and are monitored in "monitor" lifecycle stage Control measures exist and are monitored in "monitor" lifecycle stage
DMAI-Scrum <small>Blend of DMAIC and Scrum</small>	Very involved customer Process driven Higher number of unknowns Customer can be engaged independent of one another	Organization efficiency	Requirements and development of the scope follows the defined DMAIC process. Define, Measure, Analyze, they are written in business language	Team collaborates together in all aspects of creating the scope/deliverables Product owner (customer) provides the stories Constant flow of work pulled by the team - planned sequence	Team collaborates together in all aspects of creating the scope/deliverables Product owner (customer) provides the stories Constant flow of work pulled by the team - planned sequence	Team defines "how to demo" information to test each item contained in the release Control testing as stories are developed during build Sprint or during later documentation Sprint Control measures exist and are monitored in "monitor" lifecycle stage Control measures exist and are monitored in "monitor" lifecycle stage	Iterative delivery and deployment Customer involvement Control measures exist and are monitored in "monitor" lifecycle stage Control measures exist and are monitored in "monitor" lifecycle stage	Control measures exist and are monitored in "monitor" lifecycle stage Control measures exist and are monitored in "monitor" lifecycle stage Control measures exist and are monitored in "monitor" lifecycle stage Control measures exist and are monitored in "monitor" lifecycle stage
Kan-Fall <small>Blend of Kanban and Waterfall</small>	Team members are specialized, only used in certain stages Customer involvement Some amount of unknowns Stories may be passed as needed without causing significant delivery effect Infrastructure and development based	Users (does not include "why")	Requirements are written as user stories, user stories are written from the business user perspective and DO NOT include "why" Business forecast driven and are written in business user language	Team defines the lifecycle stages, most scope items need to follow Rough estimates and being identified based on relative sizing, team WIP limits and customer expectations	Recycle stages include Develop, Build, Develop Constant flow of work pulled by the team - planned sequence	Will have defined test plan to include different testing types Full and to end testing done prior to acceptance Stories typically DO NOT include "how to demo"	Big bang delivery, which may have several smaller delivery points after the supporting functionality is accepted Control measures exist and are monitored in "monitor" lifecycle stage Control measures exist and are monitored in "monitor" lifecycle stage	Stabilization, documentation, standard closing activities done at end of the project Control measures exist and are monitored in "monitor" lifecycle stage Control measures exist and are monitored in "monitor" lifecycle stage
Kan-Scrum <small>Blend of Kanban and Waterfall</small>	Team members are specialized, only used in certain stages Customer in heavily involved Higher number of unknowns Stories can be passed as needed without causing significant delivery effect	Users (does not include "why")	Requirements are written as user stories, user stories are written from the business user perspective and DO NOT include "why" Business forecast driven and are written in business user language	Team collaborates together in all aspects of creating the scope/deliverables Product owner (customer) provides the stories Constant flow of work pulled by the team - planned sequence	Team collaborates together in all aspects of creating the scope/deliverables Product owner (customer) provides the stories Constant flow of work pulled by the team - planned sequence	Team defines "how to demo" information to test each item contained in the release Control testing as stories are developed during build Sprint or during later documentation Sprint Control measures exist and are monitored in "monitor" lifecycle stage Control measures exist and are monitored in "monitor" lifecycle stage	Iterative delivery and deployment Customer involvement Control measures exist and are monitored in "monitor" lifecycle stage Control measures exist and are monitored in "monitor" lifecycle stage	Control measures exist and are monitored in "monitor" lifecycle stage Control measures exist and are monitored in "monitor" lifecycle stage Control measures exist and are monitored in "monitor" lifecycle stage Control measures exist and are monitored in "monitor" lifecycle stage



Driven by the methodology...
...not driving the methodology

- Communication/
Change
Management
- Testing
- Management
Commitment/
buy-in

Let's Practice!

Decision Tool

ntrda.me/decision

Planning

Government Data Compliance Project:

- Experienced specialized team
 - Recent Agile training
 - Customer available, when needed
 - Requirements provided
 - Loose timeline, eventual final deadline
- Level of customer participation
 - Level of requirement details needed
 - Team methodology preference/ experience
 - Primary project driver
 - Ongoing measures required post project



Execution

Government Data Compliance Project:

- Team has mixture of skills and buy-in
- Members of other high risk projects
- Potential Compliance consultants
- Discrete deliverables, delivered at will
- Some cross-dependencies



- Level of team oversight needed
- Ability to respond to change
- Level of dependencies within the project
- Level of 3rd party vendor management
- Delivery approach

Practice - Results

Planning

- 1 Level of customer participation on the project team

Full	Often	Sometimes	Intermittent
	x	x	

- 2 Level of requirement/user story details needed prior to the start of execution

High	Medium High	Medium	Low
	x		

- 3 Project team methodology preference/experience

Waterfall	DMAIC	SCRUM	Kanban
x		x	x

- 4 Primary project driver

Process	Timeline	Unfamiliar	Functionality
		x	x

- 5 Ongoing measures required post project

Required	Recommended	Optional	Rarely/ N/A
			x

Execution

- 6 Level of team oversight needed

High	Medium	Low
	x	

- 7 Ability to respond to changes to scope, timeline, and/or resources

Easy	Moderate	Difficult
	x	

- 8 Level of dependencies within the project

Many	Some	Few
	x	

- 9 Level of 3rd party vendor management (e.g. RFP)

High	Medium	Low or N/A
	x	

- 10 Delivery approach

Iterative	Constant Flow	Big Bang
x	x	

Planning Methodology

DMAIC	Waterfall	SCRUM	Kanban
2	1	2	4

Execution Methodology

Waterfall	SCRUM	Kanban
0	4	2

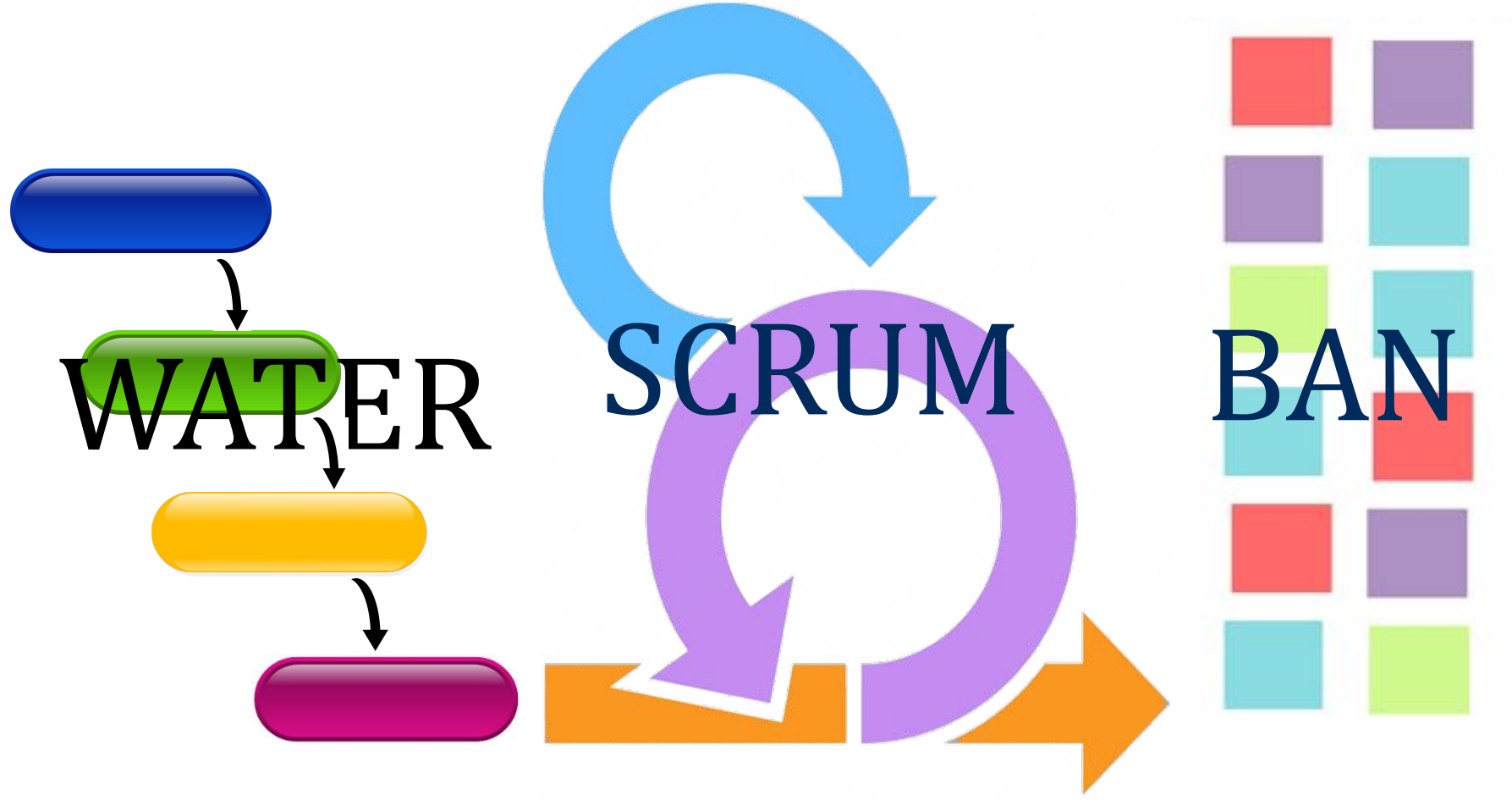
Kan-Scrum



I'D TURN BACK IF I WERE YOU!

SAFE

Secure Authentication
for Everyone





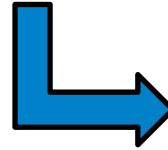
Ruby Slippers Exercise



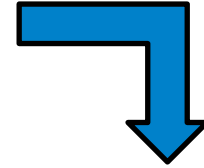


Ruby Slippers Exercise #1

SCRUM



Involved
customers get
pulled to another
project leaving the
technical team to finish
the work



SCRUM-Ban

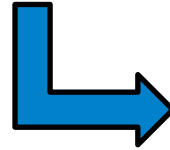
SCRUM-Fall

Cancel the
Project

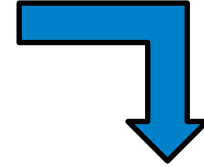


Ruby Slippers Exercise #2

Kanban



The
NEW project
sponsor wants to
see deliverable
progress on her priority
items



Kan-Fall

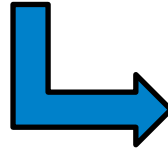
Kan-Scrum

Kan-DMAI

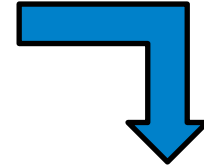


Ruby Slippers Exercise #3

SCRUM



More
resources
added due to
request for more
customer demos and a
shorter timeline



SCRUM-Fall

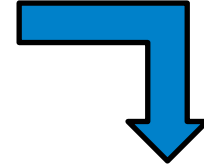
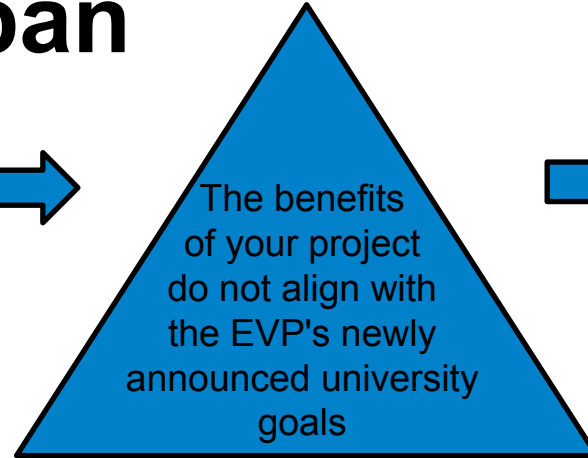
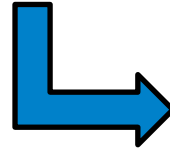
SCRUM-Ban

Stay the
Course



Ruby Slippers Exercise #4

Kanban



Kan-Fall

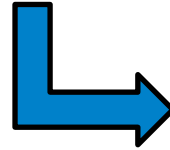
Cancel the
Project

Kan-SCRUM

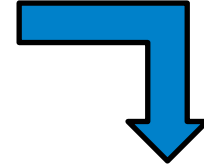


Ruby Slippers Exercise #5

Waterfall



Customers
have asked to
have products
reviewed and released
as they become available



Stay the
Course

Water-Scrum

Water-Ban



Thank you!

Sharon Hayward

Lead Project Manager
University of Notre Dame
shayward@nd.edu
574-631-2461

Michelle Johnson

Manager, Service &
Platform Management
University of Notre Dame
Michelle.Johnson@nd.edu
574-631-3740

Decision Tool
ntrda.me/decision



Go IRISH!!!



Blend Matrix
ntrda.me/blends

Special Thanks to our OIT colleagues for their contributions to this presentation.

John Hartman
Business Systems Analyst

Julie Pawlak
Lead Project Manager

Bobbi Cain
Lead QA/BA

Gina Grear
Project Manager