

LeSS



Certified LeSS Practitioner Course
Ahmad Fahmy



www.ahmadfahmy.com



@z2ahmad

ahmad@zone2consulting.com



CREDITS

Special thanks to Craig, Bas, Jurgen, Ran & Charlie!

Do not copy or share this material, or re-use for other education.

Exceptions require prior written consent of the author.

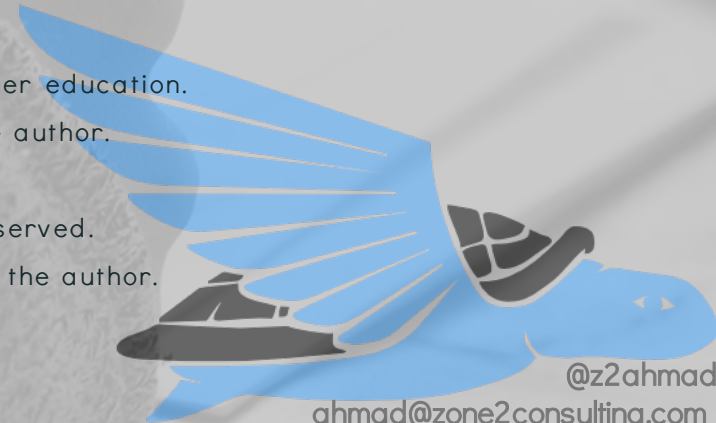
Copyright © 2016 Ahmad Fahmy, All rights reserved.

May not be reproduced without written consent of the author.



@less_works

www.ahmadfahmy.com



@z2ahmad

ahmad@zone2consulting.com



Course Mechanics

Less Powerpoint, more interaction... Lot's of war stories.

- 9:15 - 10:05 Session 1
- 10:05 - 10:15 Break
- 10:15 - 11:05 Session 2
- 11:05 - 11:15 Break
- 11:15 - 12:10 Session 3
- 12:10 - 1:00 Lunch
- 1:00 - 1:50 Session 4
- 1:50 - 2:00 Break
- 2:00 - 2:50 Session 4
- 2:50 - 3:00 Break
- 3:00 - 3:50 Session 5
- 3:50 - 4:00 Break
- 4:00 - 5:00 Session 6
- 5:00 - 5:30 Check-in in a QA

- No Phones
- No Laptops
- Questions, questions, questions
- Q&A Wall



@less_works

www.ahmadfahmy.com



@z2ahmad

ahmad@zone2consulting.com



AGENDA

Focus on the “*practitioner*” part of Certified LeSS practitioner

- Scaling
- The story of LeSS
- LeSS Framework
- Less Principles
- Systems Thinking
- Lean Thinking
- Roles
- Management
- Meetings
- Artifacts
- Coordination
- Technical Excellence
- LeSS adoption
- Conclusion & Next steps



@less_works

www.ahmadfahmy.com



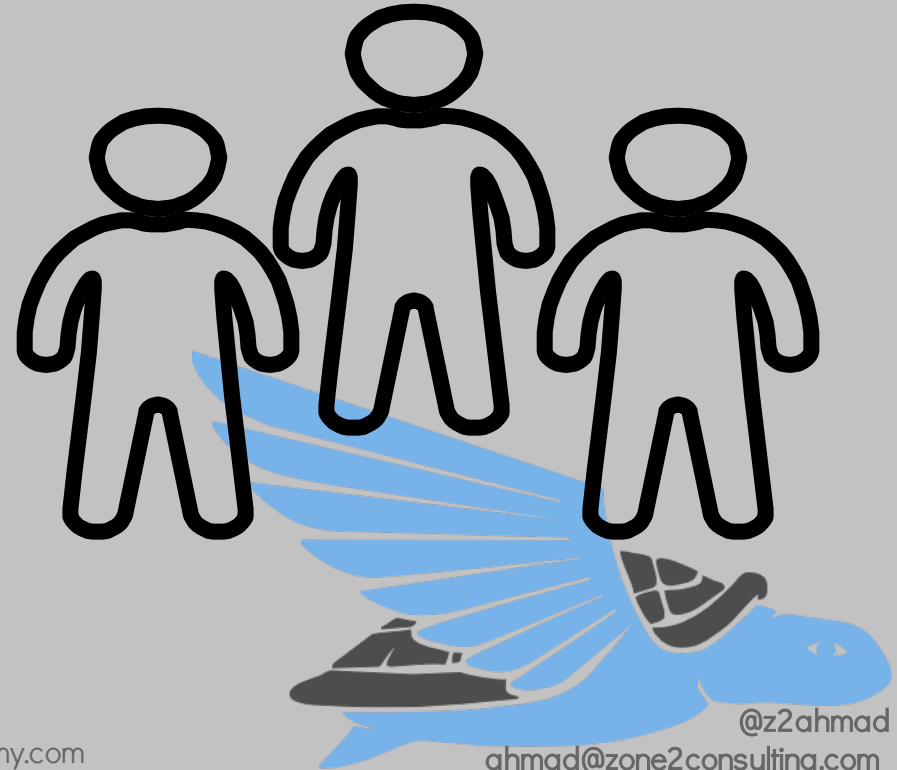
@z2ahmad

ahmad@zone2consulting.com



Sociogram

- LeSS, Scrum familiarity
- domains (e.g., banking, telecom, ...)
- role



HELLO 😊

I am Ahmad

- Started my career as a full time developer at 18
- My Story
- Working on LeSS transformations since 2010
- LeSS transformations at large banks to small product companies
- JP Morgan Chase, Bank Of America, ION trading, Tripple Point Technologies



@less_works

www.ahmadfahmy.com



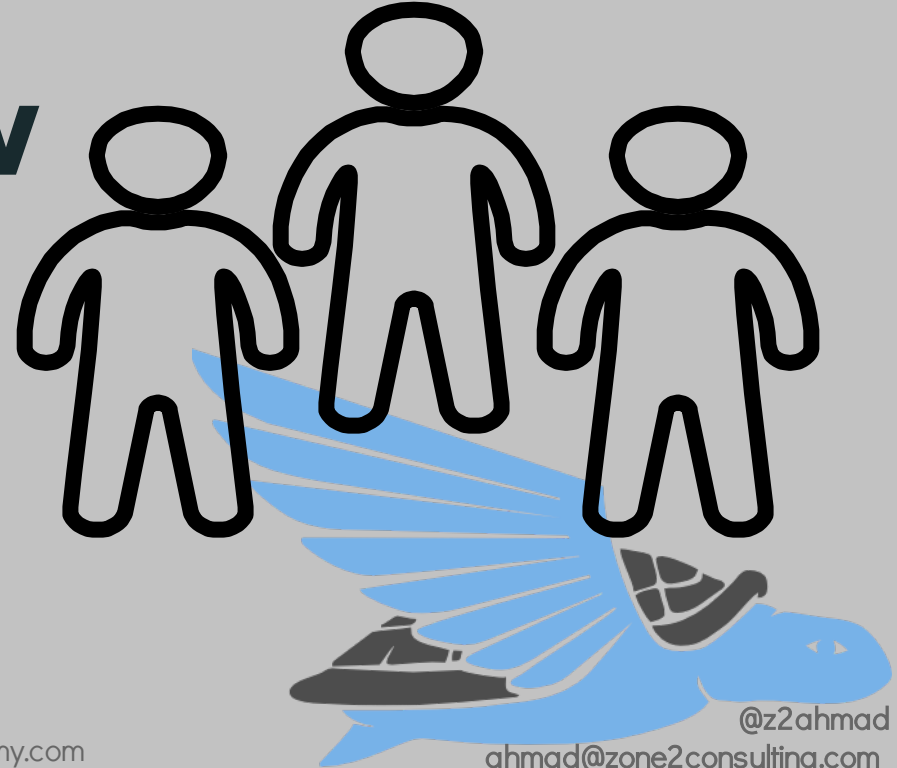
@z2ahmad

ahmad@zone2consulting.com



Get to know each other

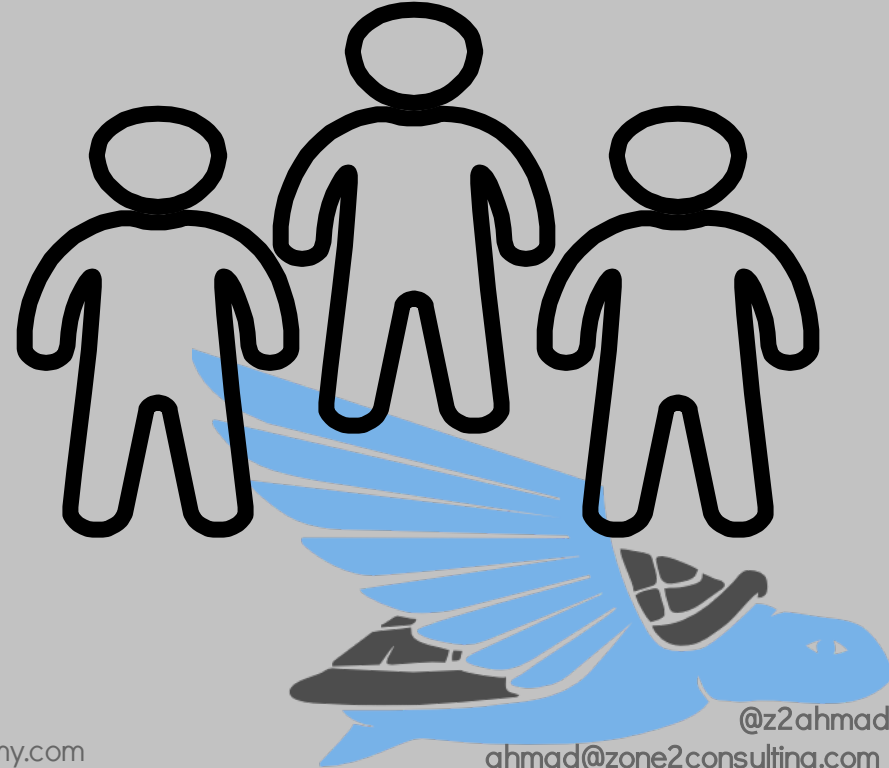
- **briefly** introduce each other
- Who are you?
- Something memorable

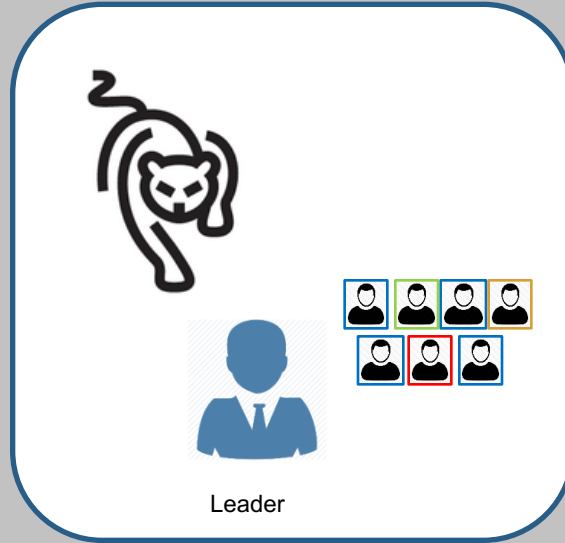




The one thing...

- ❑ Look back at your career, visualize a successful project.
- ❑ Write down what was the single biggest factor in the success of that project.





In the Beginning...

www.ahmadfahmy.com





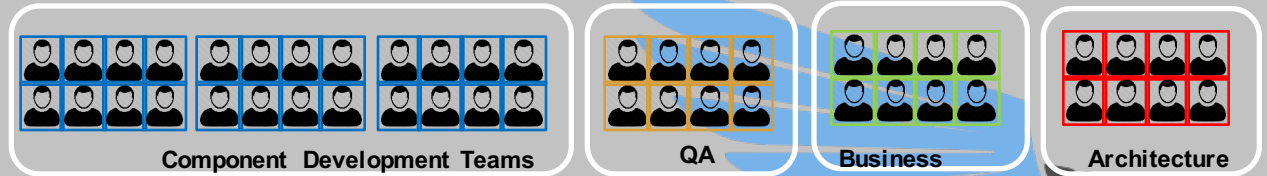
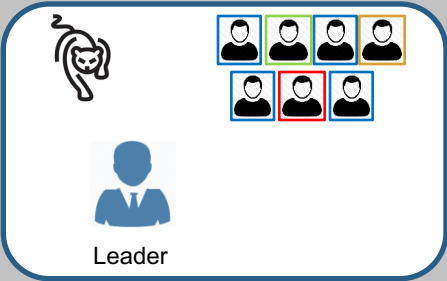
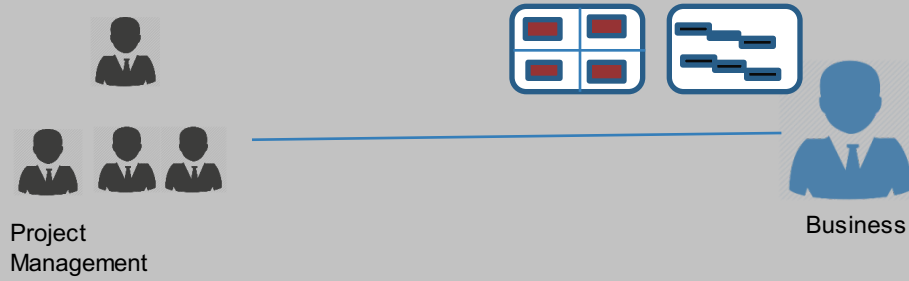


Purpose

Autonomy

Mastery

What it was like...



How we got here....

BB

Across all sized projects, agile projects are 350% more likely to be successful. This difference is minimal when running small projects - 32%. But at the huge project end of the spectrum, agile projects are 600% more likely to be successful.

-2015 Chaos Report



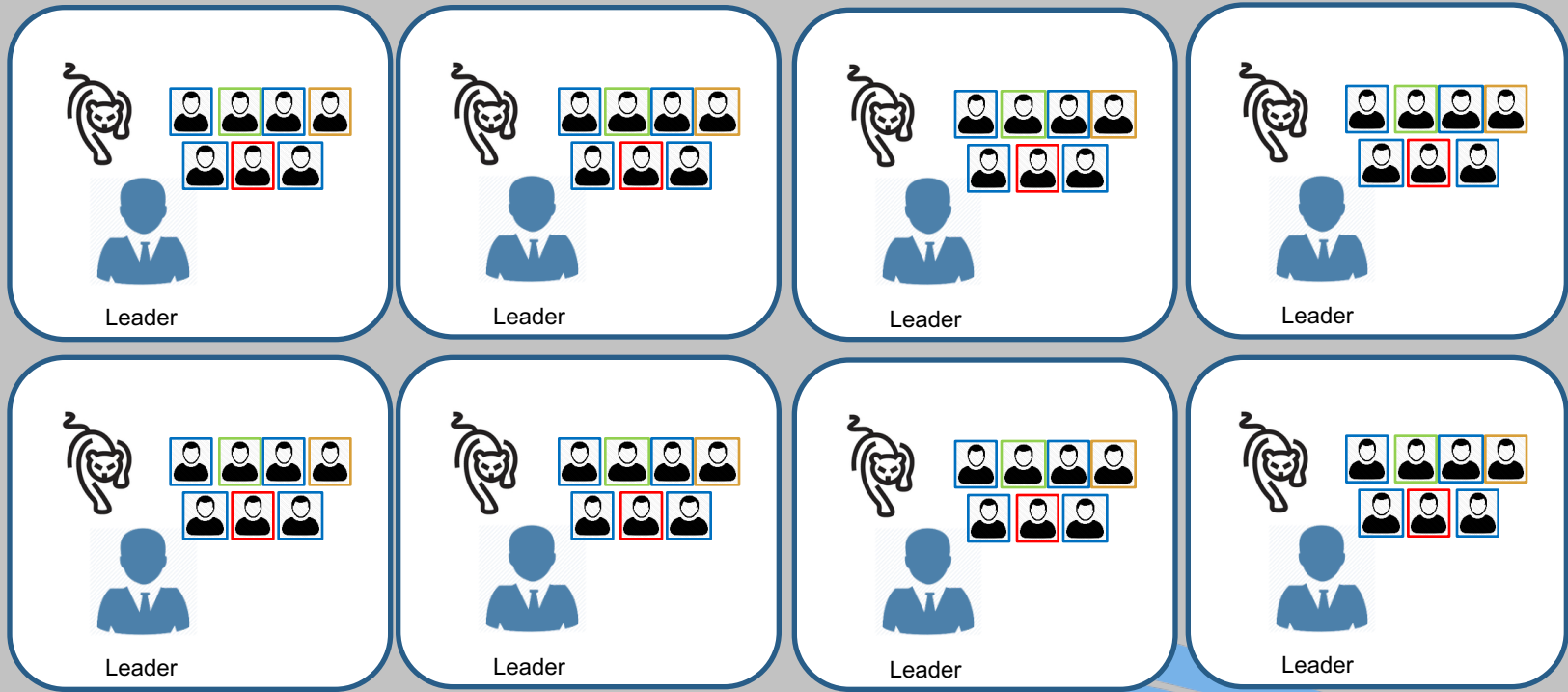
@less_works

www.ahmadfahmy.com



@z2ahmad

ahmad@zone2consulting.com



What are we trying to do?



BB

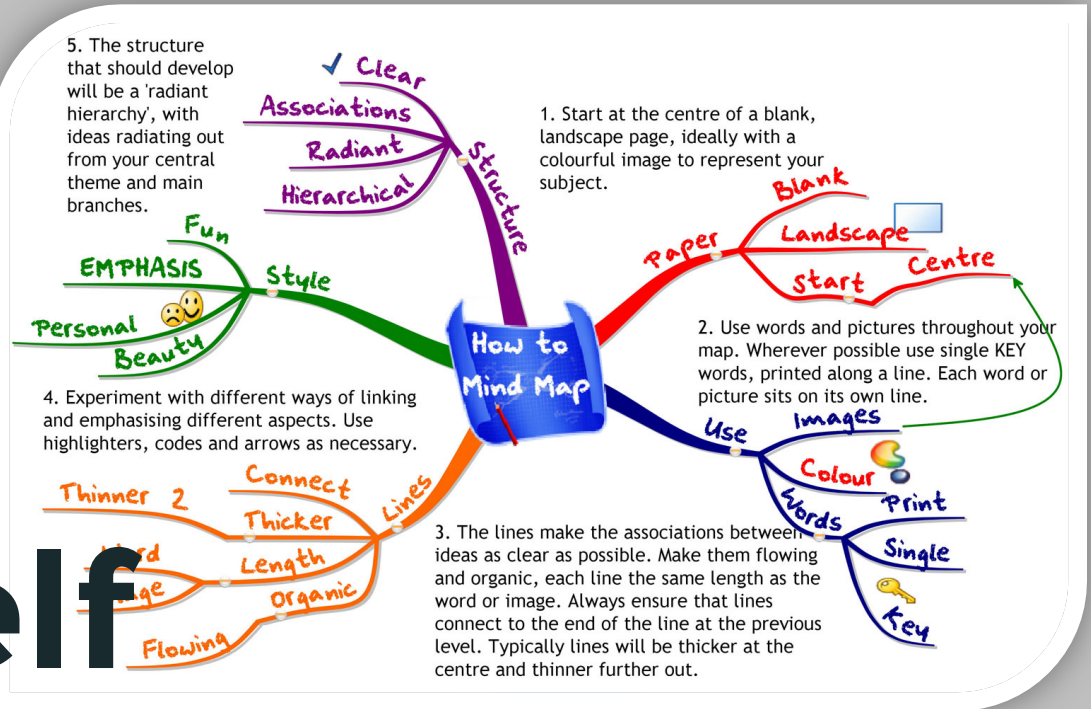
If you had it to do all over again, what would you do differently? “find the ten best people and write the entire thing themselves.”

-Sage Program Director





Make yourself brilliant



1

What is LeSS

Demystifying LeSS



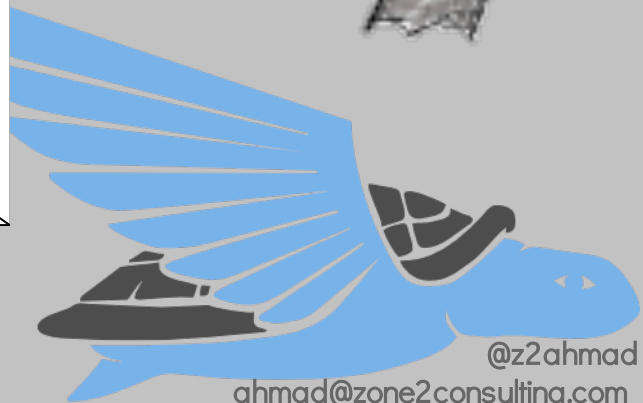
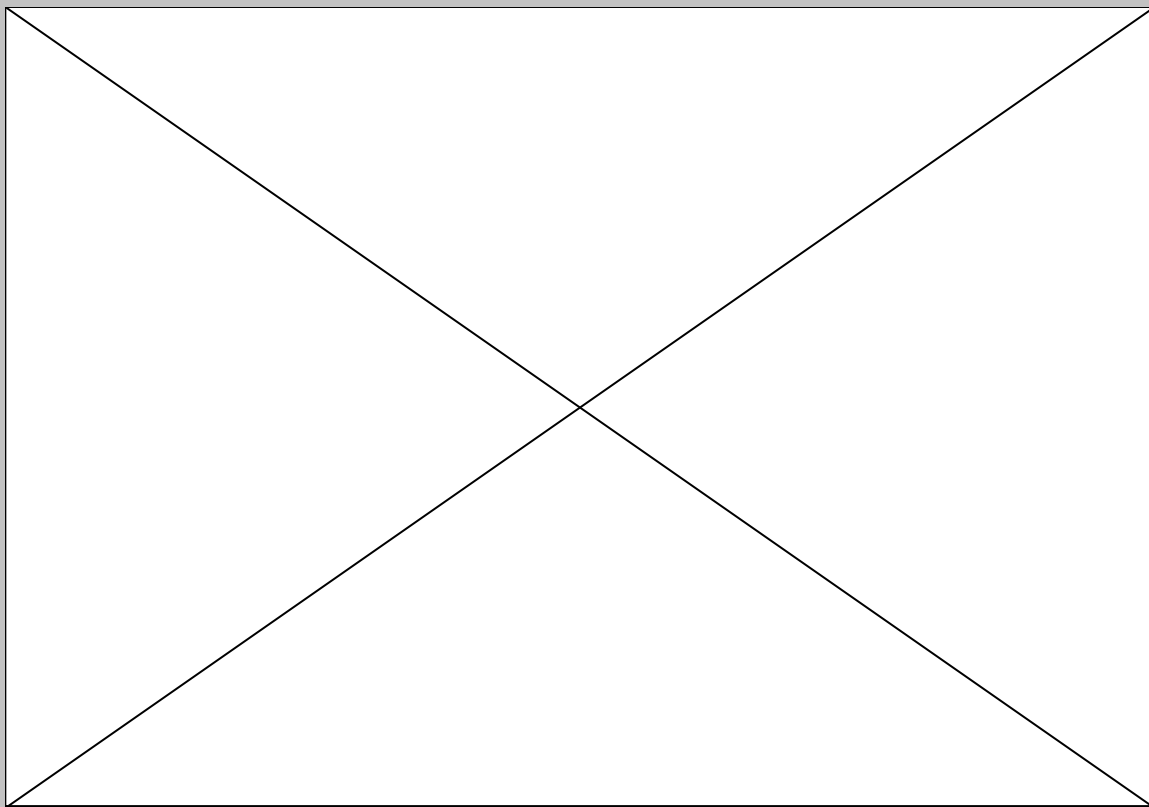
@less_works

www.ahmadfahmy.com



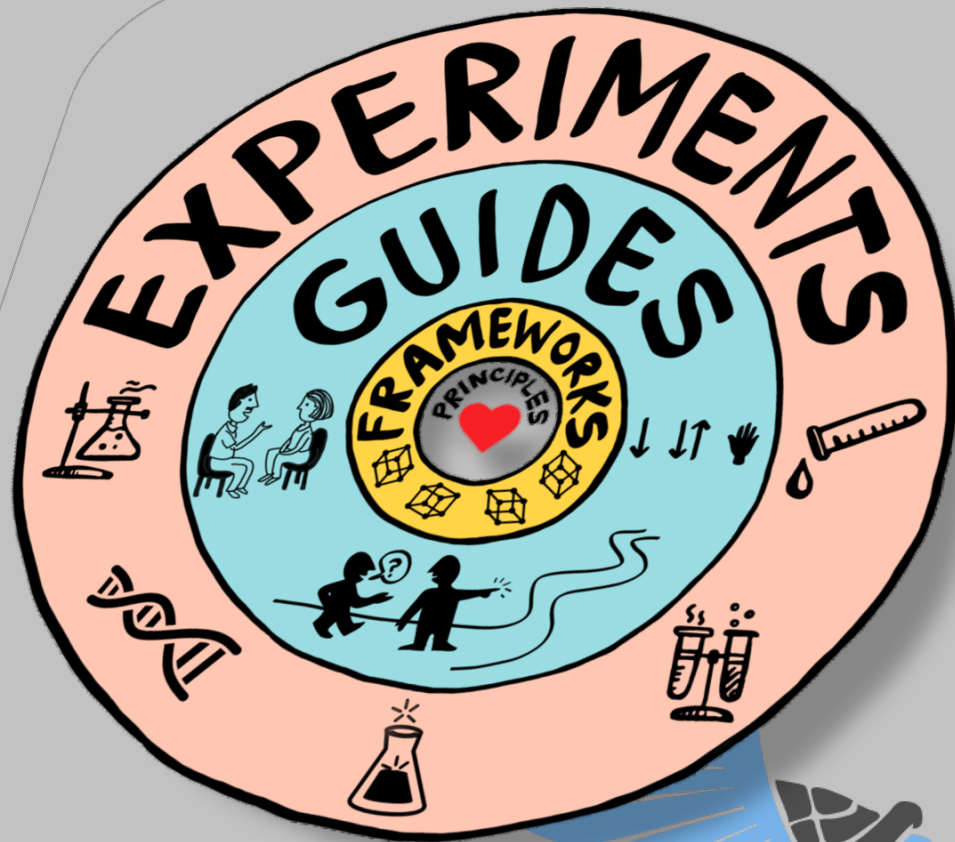
@z2ahmad

ahmad@zone2consulting.com





A Picture of LeSS





LeSS hits the sweet spot between abstract principles and concrete practices.

LeSS needs to be simple

LeSS is Scrum Scaled

Scaled up instead of tailored
down



@less_works

www.ahmadfahmy.com



@z2ahmad

ahmad@zone2consulting.com

Less Framework

2-8 Teams



@less_works

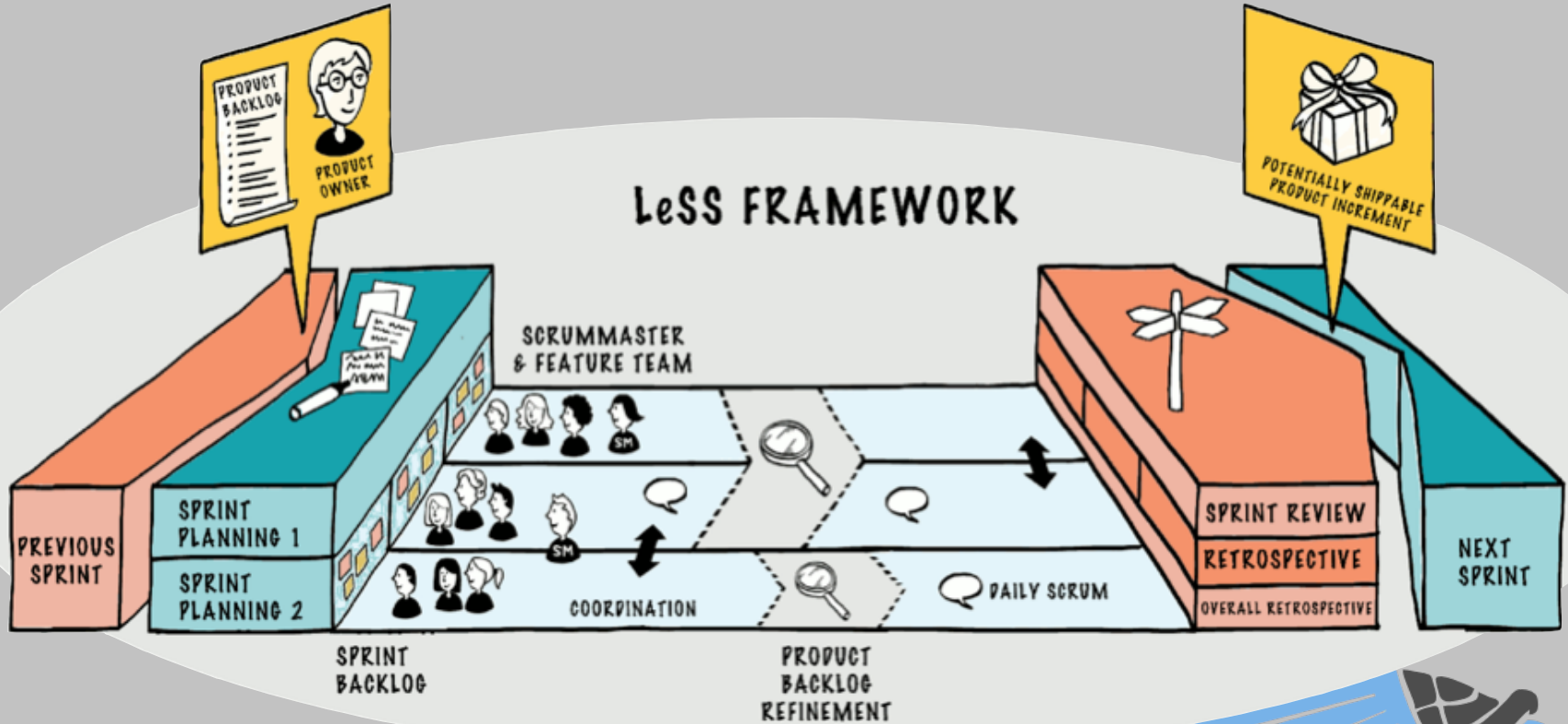
www.ahmadfahmy.com



@z2ahmad

ahmad@zone2consulting.com

LeSS FRAMEWORK





LeSS Structure

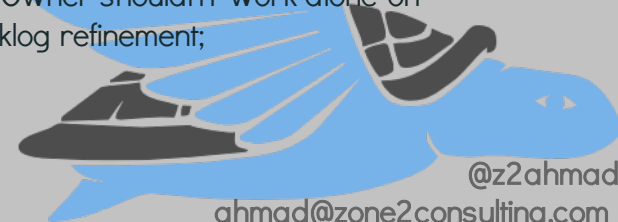
- Each team is (1) self-managing, (2) cross-functional, (3) co-located, and (4) long-lived.
- A ScrumMaster is a dedicated full-time role.
- One ScrumMaster can serve 1-3 teams.
- Managers work to improving the value-delivering capability of the product development system.
- For the product group, establish the complete LeSS structure “at the start”; this is vital for a LeSS adoption.
- Teams are the building blocks of the organization
- ScrumMasters are responsible for a well-working LeSS adoption.
- For the larger organization beyond the product group, adopt LeSS evolutionarily using Go and See to create an organization where experimentation and improvement is the norm.
- The majority of the teams are customer-focused feature teams.





LeSS Product

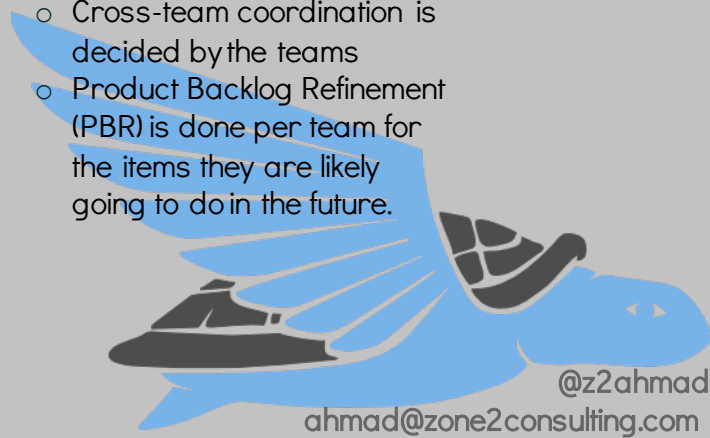
- There is one Product Owner and one Product Backlog for the complete shippable product
- All prioritization goes through the Product Owner,
- One shared Definition of Done for the whole product
- Each team can have their own expanded Definition of Done.
- The definition of product should be as broad and end-user/customer centric as is practical.
- The perfection goal is to improve the Definition of Done so that it results in a shippable product each Sprint (or even more frequently).
- The Product Owner shouldn't work alone on Product Backlog refinement;



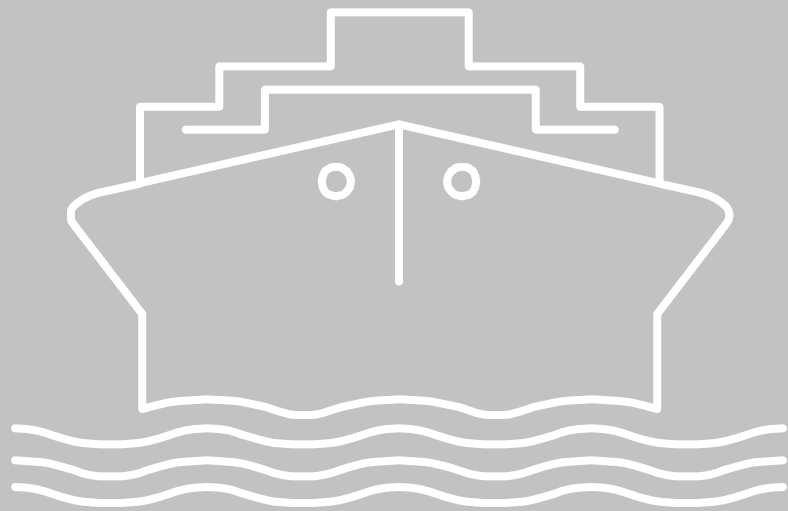


LeSS Sprint

- There is one product-level Sprint
- Sprint Planning consists of two parts
- Sprint Planning Part One is attended by the Product Owner and Teams or Team representatives.
- Each Team has their own Sprint Backlog.
- Sprint Planning Part Two is for Teams to decide how they will do the selected items
- Each Team has their own Daily Scrum
- There is one product Sprint Review; it is common for all teams
- An Overall Retrospective is held after the Team Retrospectives to discuss cross-team and system-wide issues, and create improvement experiments.
- Each Team has their own Sprint Retrospective.
- Cross-team coordination is decided by the teams
- Product Backlog Refinement (PBR) is done per team for the items they are likely going to do in the future.



Less



HUGE

>8 Teams



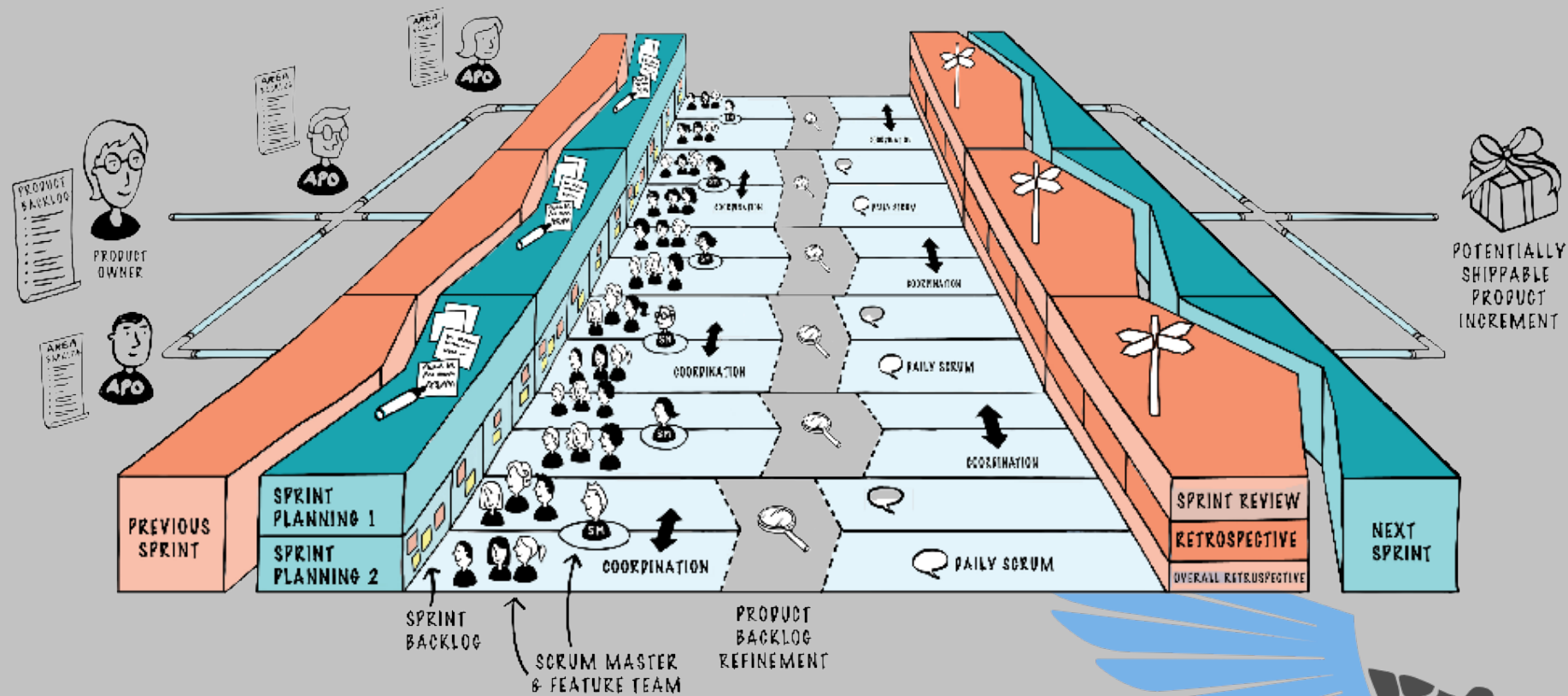
@less_works

www.ahmadfahmy.com



@z2ahmad

ahmad@zone2consulting.com





LeSS (h) Structure

- Each Team specializes in one Requirement Area.
- Each Requirement Area has one Area Product Owner.
- Each Requirement Area has between “4-8” teams.
- Customer requirements that are strongly related from a customer perspective are grouped in Requirement Areas.
- LeSS Huge adoptions, including the structural changes, are done with an evolutionary incremental approach.
- Remember each day: LeSS Huge adoptions take months or years, infinite patience, and sense of humor.





LeSS (h) Product

- Each Requirement Area has one Area Product Owner.
- One (overall) Product Owner is responsible for product-wide prioritization
- Area Product Owners act as Product Owners towards their teams.
- There is one Product Backlog; every item in it belongs to exactly one Requirement Area.
- There is one Area Product Backlog per Requirement Area.



@less_works

www.ahmadfahmy.com



@z2ahmad

ahmad@zone2consulting.com



LeSS (h) Sprint

- **There is one product-level Sprint**
- All Sprint LeSS rules apply for each Requirement Area.
- The Product Owner and Area Product Owners synchronize frequently.
- A Sprint Review is held per Requirement Area.
- An Overall Retrospective is held per Requirement Area.



LeSS

Principles



@less_works

www.ahmadfahmy.com



@z2ahmad

ahmad@zone2consulting.com

BB

The LeSS rules define the LeSS Framework. But the rules are minimalistic and do not give answers as to how best to apply LeSS in your specific context. The LeSS principles provide the basis for making those decisions.



@less_works

www.ahmadfahmy.com



@z2ahmad

ahmad@zone2consulting.com



QUEUEING THEORY

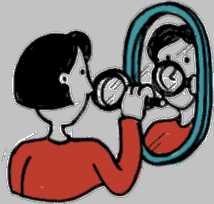


**LARGE-SCALE
SCRUM IS SCRUM**

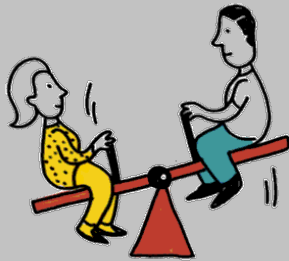
TRANSPARENCY



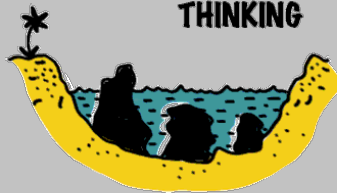
**MORE
WITH LESS**



**EMPIRICAL
PROCESS CONTROL**



**SYSTEMS
THINKING**



**LEAN
THINKING**

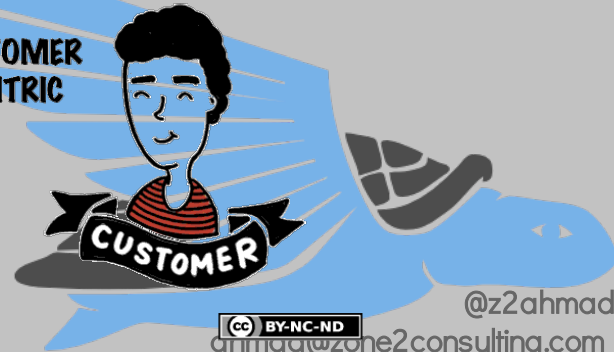


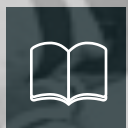
**CONTINUOUS IMPROVEMENT
TOWARDS PERFECTION**



**WHOLE
PRODUCT
FOCUS**

**CUSTOMER
CENTRIC**





LARGE SCALE SCRUM *IS* SCRUM



BB

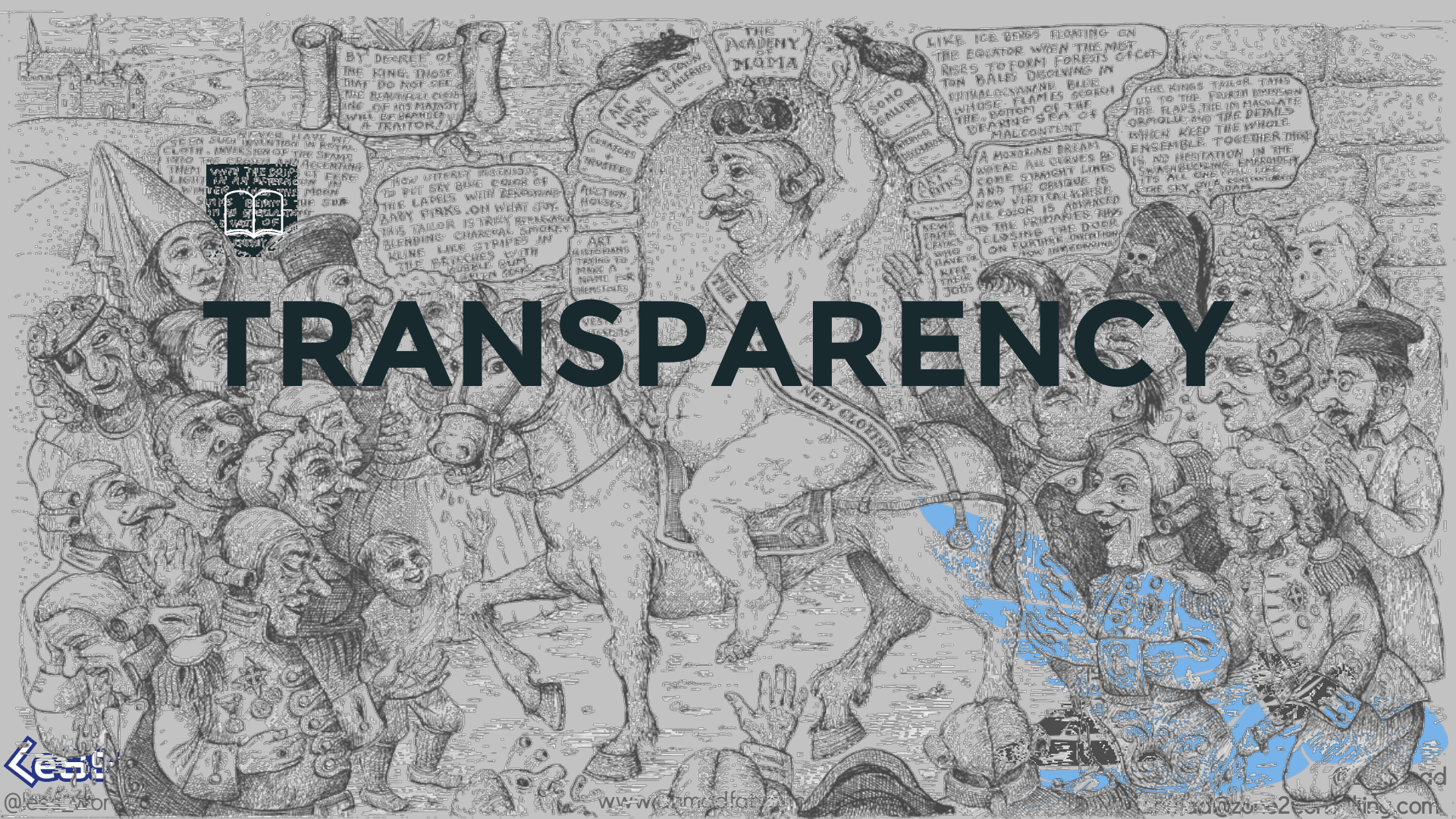
There will be no Scrum Release 2.0...Why not? Because the point of Scrum is not to solve [specific problems of development]... Scrum unearths the problems caused by the complexity and lets the organization solve them, one by one, over and over again. [Schwaber07]



MORE WITH LeSS



TRANSPARENCY





WHOLE PRODUCT FOCUS



Quality ←

Customer ←

Efficiency

CUSTOMER

CENTRIC

Service

Reliability



CONTINUOUS IMPROVEMENT TOWARDS PERFECTION



EMPIRICAL PROCESS CONTROLL



QUEUING THEORY



LEAN THINKING





SYSTEMS THINKING





LET'S REVIEW SOME CONCEPTS

LeSS Framework

LeSS Huge Framework

Principles



@less_works

www.ahmadfahmy.com



@z2ahmad

ahmad@zone2consulting.com

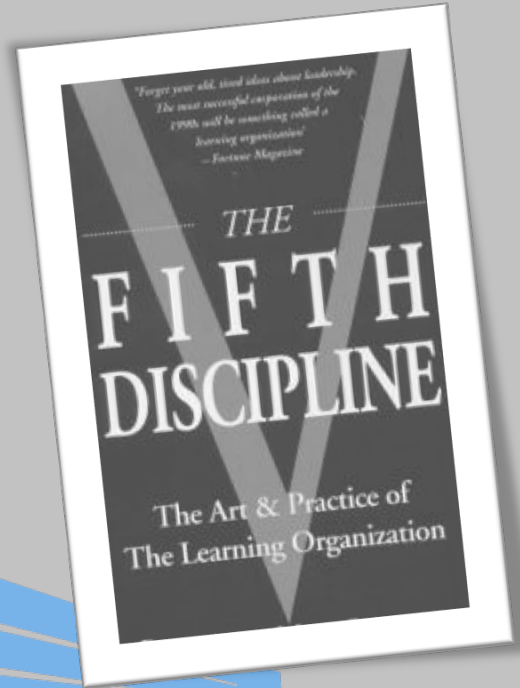
SHARED
VISION

MENTAL
MODELS

PERSONAL
MASTERY

TEAM
LEARNING

SYSTEMS
THINKING



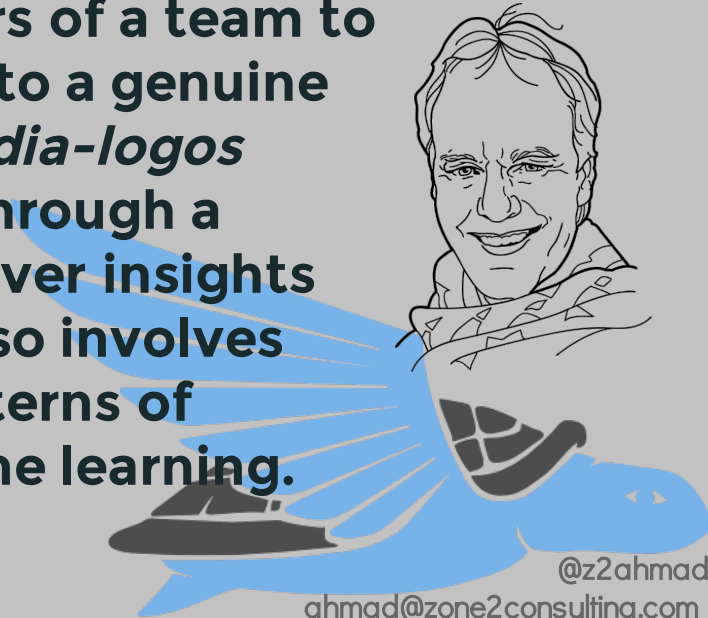
BB

When you ask people about what it is like being part of a great team, what is most striking is the meaningfulness of the experience. People talk about being part of something larger than themselves, of being connected, of being generative. It become quite clear that, for many, their experiences as part of truly great teams stand out as singular periods of life lived to the fullest. Some spend the rest of their lives looking for ways to recapture that spirit. (Senge 1990: 13)



BB

The discipline of team learning starts with 'dialogue', the capacity of members of a team to suspend assumptions and enter into a genuine 'thinking together'. To the Greeks *dia-logos* meant a free-flowing if meaning through a group, allowing the group to discover insights not attainable individually.... [It] also involves learning how to recognize the patterns of interaction in teams that undermine learning. (Senge 1990: 10)



1. I am my position.

- We often use our jobs as a proxy for our identity, often failing to understand the purpose of what we are doing for our organization; or we perceive ourselves as having little power and no need to take responsibility.

2. The enemy is out there

- We fail to understand that external and internal problems are part of the same overall system.

3. Illusion of taking charge.

- We often confuse reactive action as proactive when dealing with problems by focusing on outside threats instead of first determining how we contribute to the problem.

4. Fixation on events.

- We are too focused on the short term, which prevents us from seeing long-term patterns of change that are the cause of the immediate events.

5. Delusion of learning from experience.

- People seldom directly experience consequences of their decisions

6. Myth of the management team.

- We tend not to work together but rather fight over turfs and avoid doing anything that risks our looking bad.

6 Learning Disabilities

BB

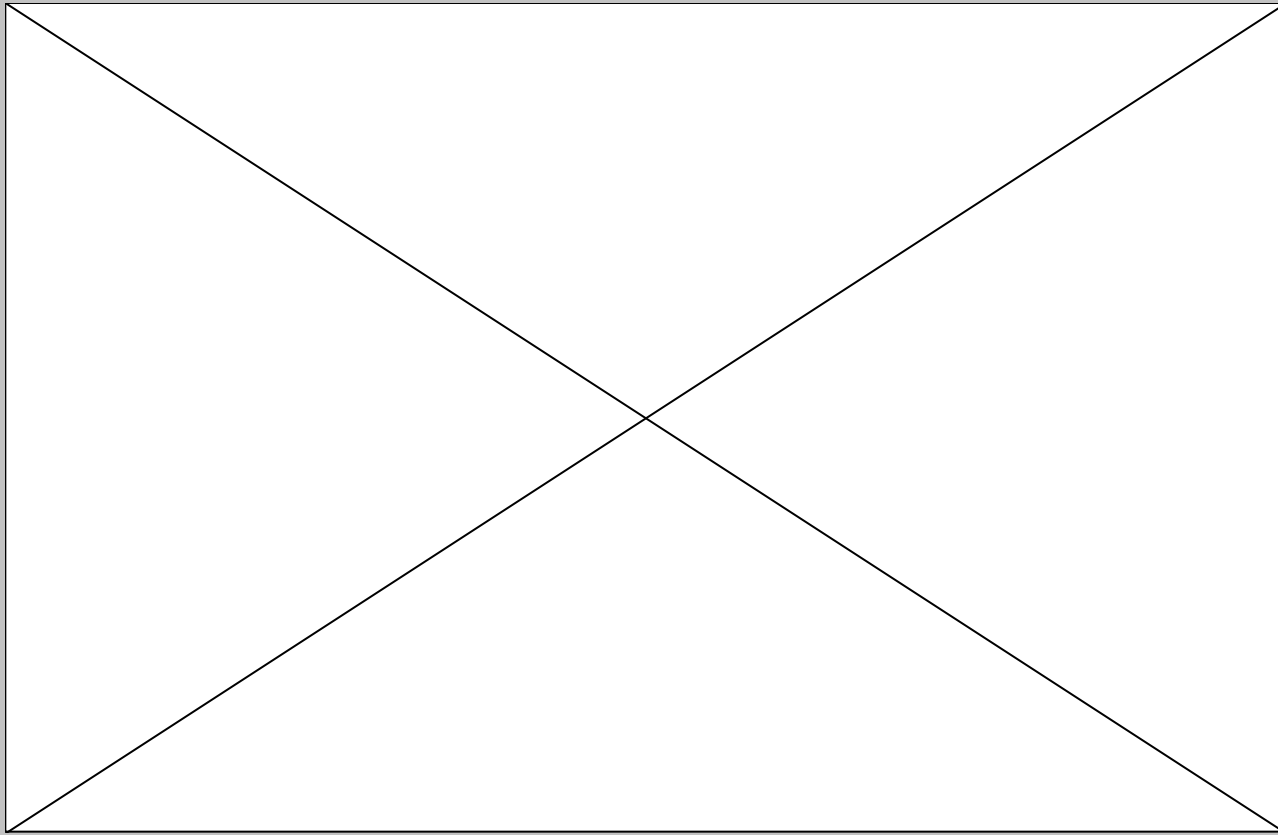
Systems thinking is a discipline for seeing whole rather than parts, for seeing patterns of change rather than static snapshots, and for understanding and subtle interconnectedness that gives living systems their unique character."

Peter Senge



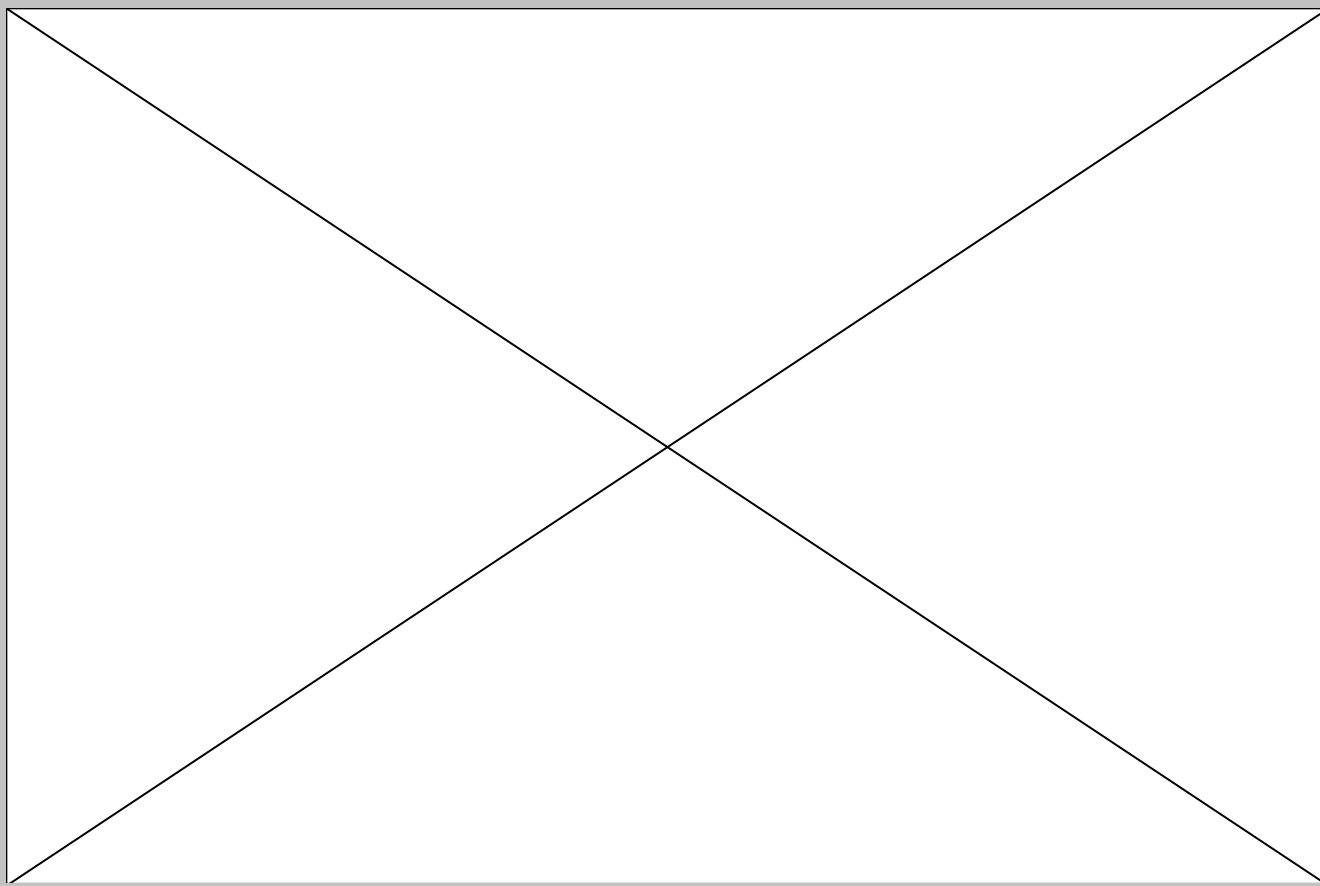
DELAY	<ul style="list-style-type: none"> • There is always a delay between the execution of actions and the final (long-term) results.
LIMITED GROWTH	<ul style="list-style-type: none"> • A pattern of limited growth is the result of focusing on improving activities which focus on improving growth accelerating factors instead of reducing growth limiting factors.
MOVING THE PROBLEM	<ul style="list-style-type: none"> • Moving the problem instead of solving it. This is what happens when only symptoms of the problem are addressed and not the root cause, The problem can then re-occur, in the same form but also in another department.
Deteriorating Goals	<ul style="list-style-type: none"> • Deteriorating Goals when situations get tough. Goals are set aside due to a crisis or because of any other reason. This is simply not acceptable. The vision and its goals give direction to the company, especially in these difficult times!
An escalation loop	<ul style="list-style-type: none"> • An escalation loop is a loop in which actors influence one another with a lose-lose situation as outcome. An example is a price-war between supermarkets, where multiple competitors eventually fight one another on being the cheapest, and none of them ends up with profit in the end. According to Senge, one should only encourage a culture in which win-win situations are created.
Success to the successful	<ul style="list-style-type: none"> • Success to the successful is the archetype in which resources are allocated to the most successful activity which makes the unsuccessful ones even more unsuccessful because they receive fewer resources. This is not necessarily the best policy for the long term.
The politics to receive resources	<ul style="list-style-type: none"> • The politics to receive resources (for instance the budgeting game) is a situation where departments make up and alter numbers to receive more resources for their department instead of being able to see the scope of the entire organization and act accordingly.
Solutions which do not solve	<ul style="list-style-type: none"> • Solutions which do not solve, is a situation where short term positive results lead to long term losses. For instance reducing Preventative maintenance on machines in a factory.
Growth and underinvestment	<ul style="list-style-type: none"> • Growth and underinvestment, is the trap where investing does not seem necessary because all is well at the moment. Not investing today, however, might lead to a lost opportunity for growth in the future because of a lack of skills or capacity.

9 Archetypes

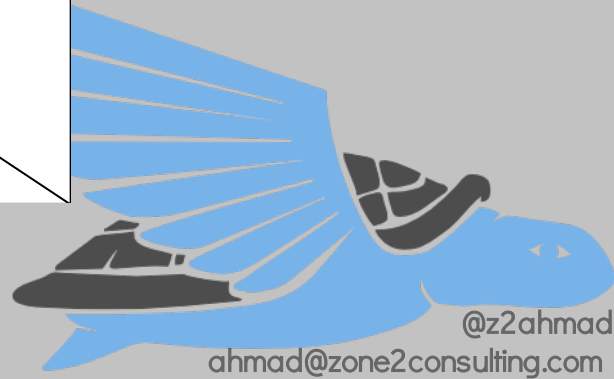


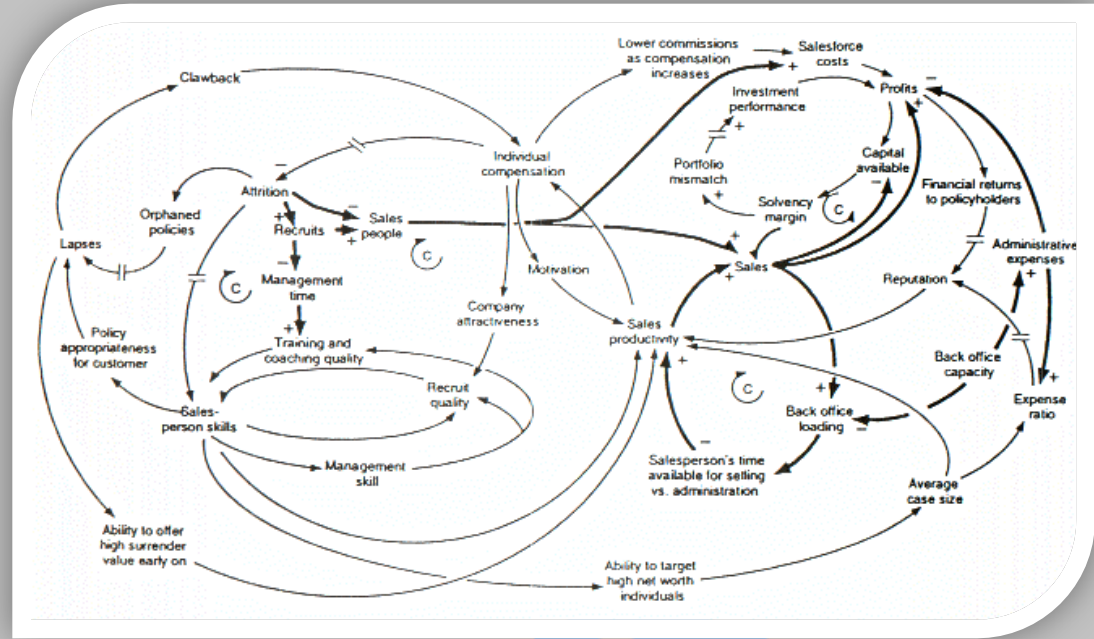
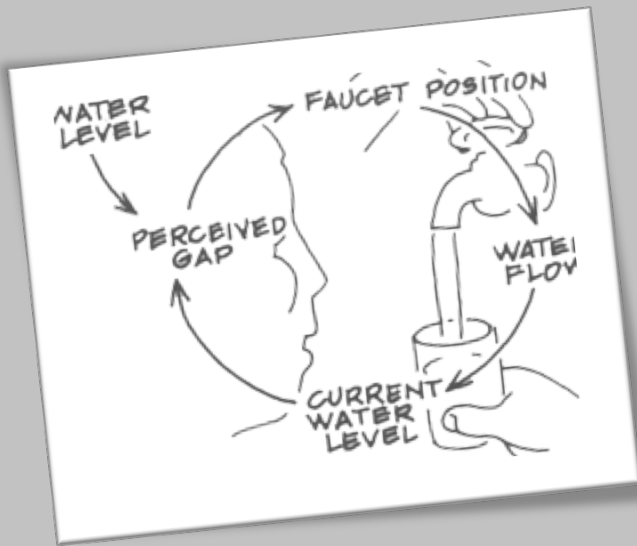
A systemic failure

www.ahmadfahmy.com



Who sees the system





- All models are wrong, but some are useful
- Model for understanding, not truth
- Model to have a conversation

Remember!

www.ahmadfahmy.com



Sustainable shortest lead time, best quality and value (to people and society), most customer delight, lowest cost, high morale, safety

**Respect
for People**

- don't trouble your 'customer'
- "develop people, then build products"
- no wasteful work
- teams & individuals evolve their own practices and improvements
- build partners with stable relationships, trust, and coaching in lean thinking
- develop teams

Product Development

- long-term great engineers
- mentoring from manager-engineer-teacher
- cadence
- cross-functional
- team room + visual mgmt
- entrepreneurial chief engineer/product mgr
- set-based concurrent dev
- create more knowledge

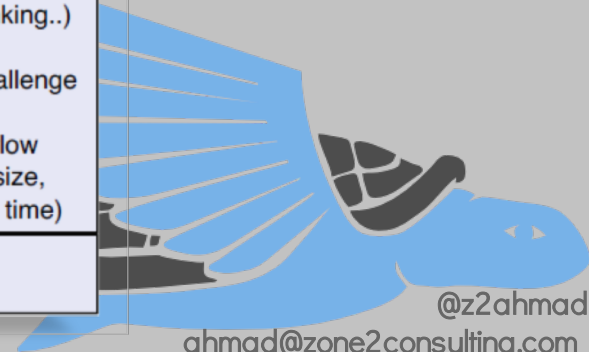
14 Principles

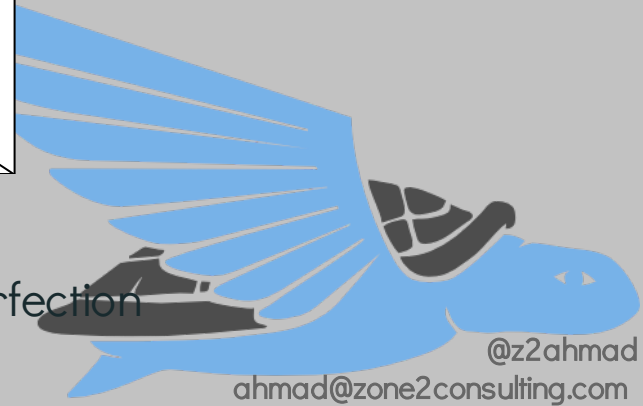
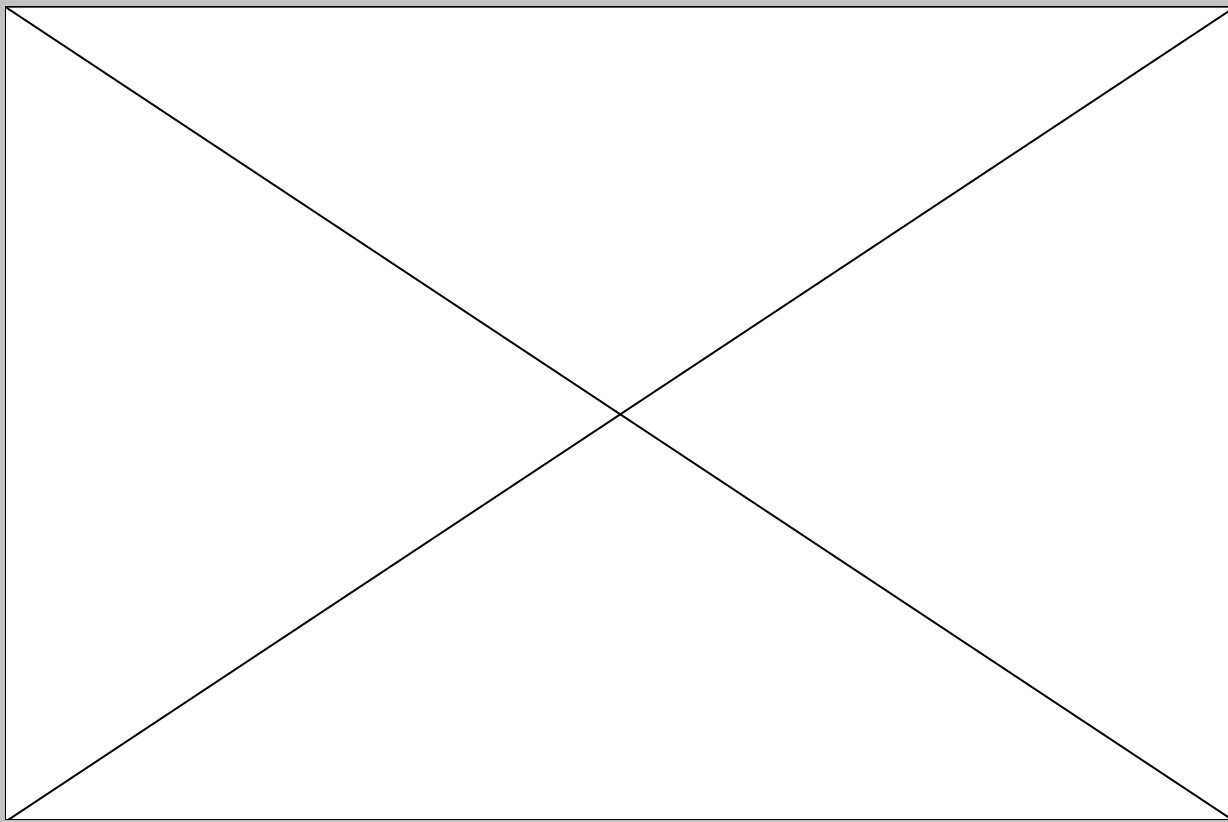
long-term, flow, pull, less variability & overburden, Stop & Fix, master norms, simple visual mgmt, good tech, leader-teachers from within, develop exceptional people, help partners be lean, Go See, consensus, reflection & kaizen

**Continuous
Improvement**

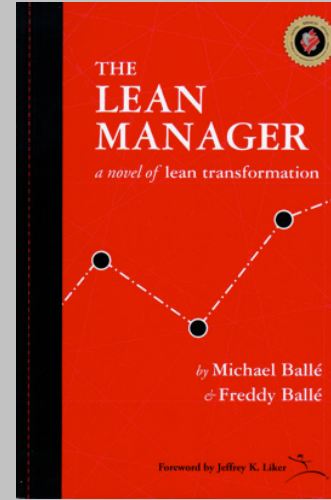
- Go See
- kaizen
- spread knowledge
- small, relentless
- retrospectives
- 5 Whys
- eyes for waste
 - * variability, overburden, NVA ... (handoff, WIP, info scatter, delay, multi-tasking, defects, wishful thinking..)
- perfection challenge
- work toward flow (lower batch size, Q size, cycle time)

Management applies and teaches lean thinking, and bases decisions on this long-term philosophy





- truly understand problems by going to Gemba
- focus on improvement for improvement sake (kaizen)
- teach problem solving
- NOT micro-manage, collect status, or report





Spartans

Spartans

SRINIVAS BEVARA

044

043

DIOVMAY BAGLEKAR







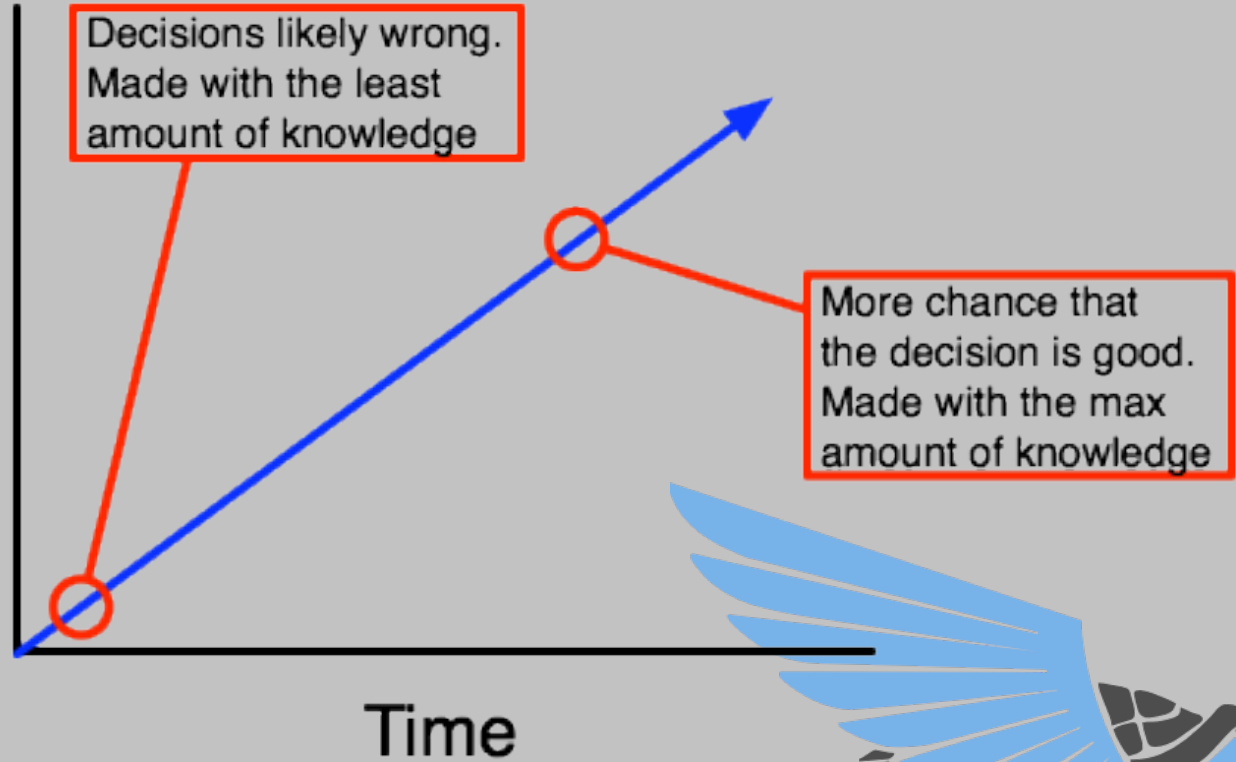
Apple
Mango
Banana
Alma Masala Grill
Alma Masala Cheese Grill
Veg. Russian Slew
Butter Milk
Bun Muska Par





Watch the baton not the players

Customer Understanding & Product Knowledge



Defer commitment



1. **Over-production**—of intermediate, WIP, or finished things; sooner, faster, greater than demand
2. **Inventory**—intermediate, WIP, or finished things
3. **Over-processing**—& extra processes, rediscovery
4. **Conveyance**—& handoff
5. **Motion**—& task switching
6. **Waiting**—& delay
7. **Defects & finding/correcting**—tasks to find & correct: test, inspect, review, modify
8. **Not using people's full potential**—working to title, not multi-skilling
9. **Knowledge/information scatter/loss**—& connection to handoff & inventory & rediscovery; communication barriers: indirection, 1-way flows
10. **Wishful thinking**—
[design/spec/estimate] is correct, learning & feedback is only mildly important, what said=what heard, ...

Lean waste in Product Development



@less_works

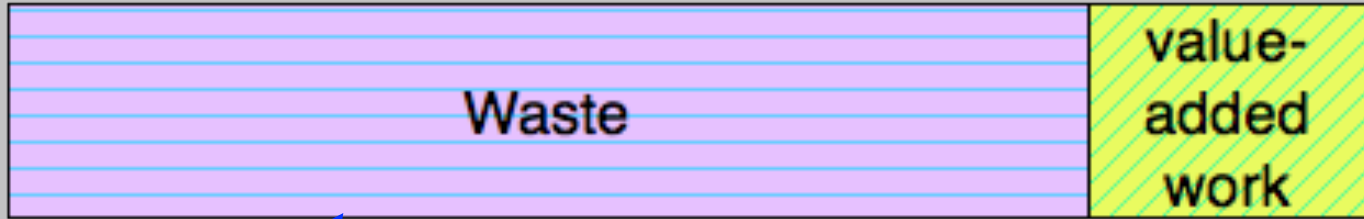
www.ahmadfahmy.com



@z2ahmad

ahmad@zone2consulting.com

Total Time Spend of an Engineer

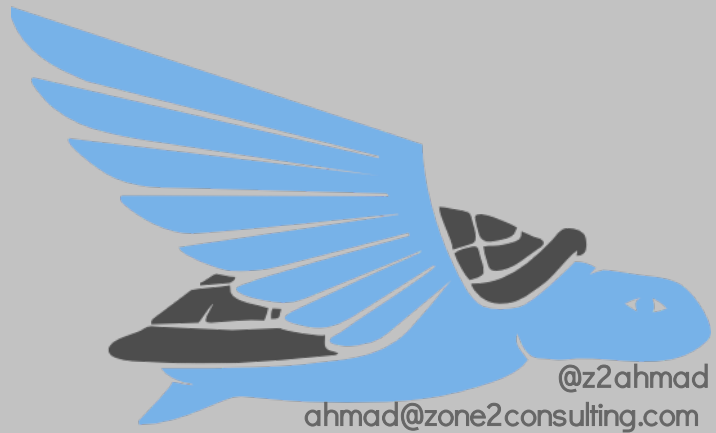


Opportunity

Waste management

Traditional
Improvement
Effort

Time

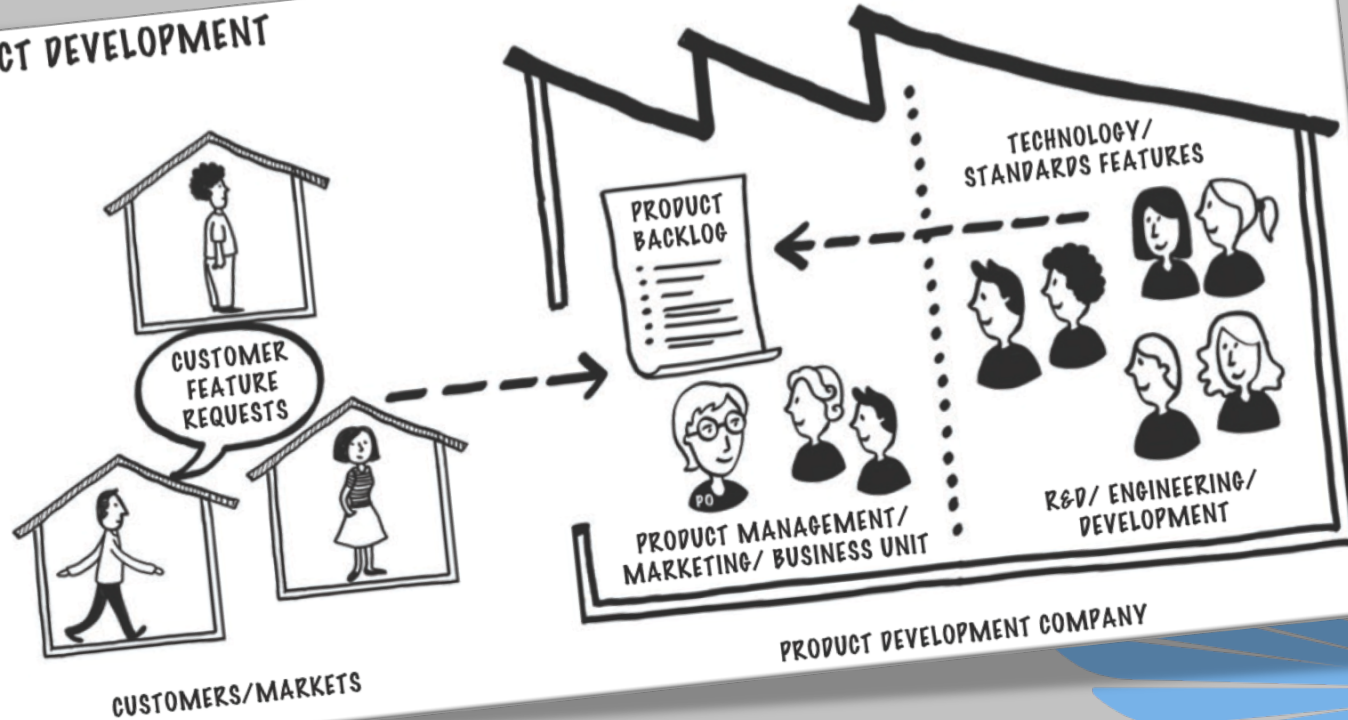


A black and white photograph of a man with dark hair, wearing a horizontally striped sweater, sitting in a chair. He is holding a cigarette in his right hand and resting his left hand on his knee. He is looking thoughtfully to his right. The background is a cluttered office space with shelves filled with books, papers, and various small objects. Two vintage computer monitors are visible behind him, one of which is displaying a bright image. The overall atmosphere is one of a busy, creative, and somewhat chaotic work environment.

Product Owner

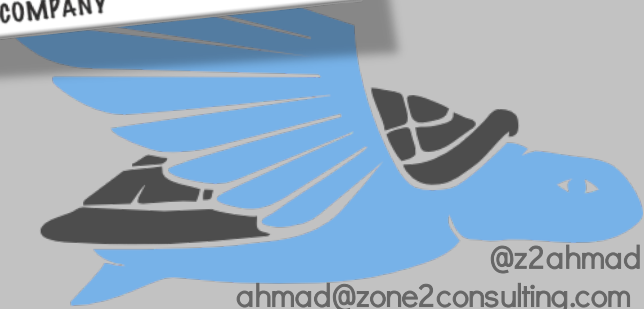
[Business] is used to “throwing the project over the wall” and holding engineering/development responsible for meeting needs. Scrum puts this responsibility back on the Product Owner and customers through the inspect and adapt and the Sprint Review.

PRODUCT DEVELOPMENT

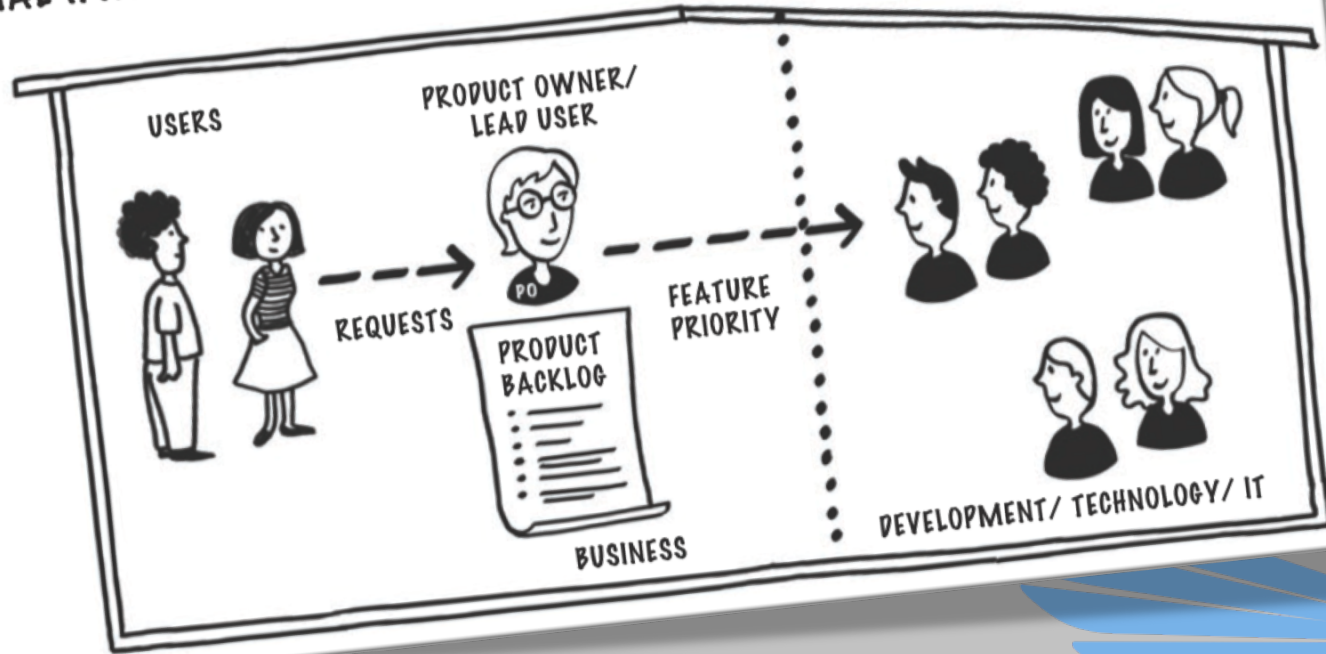


Product Development

www.ahmadfahmy.com

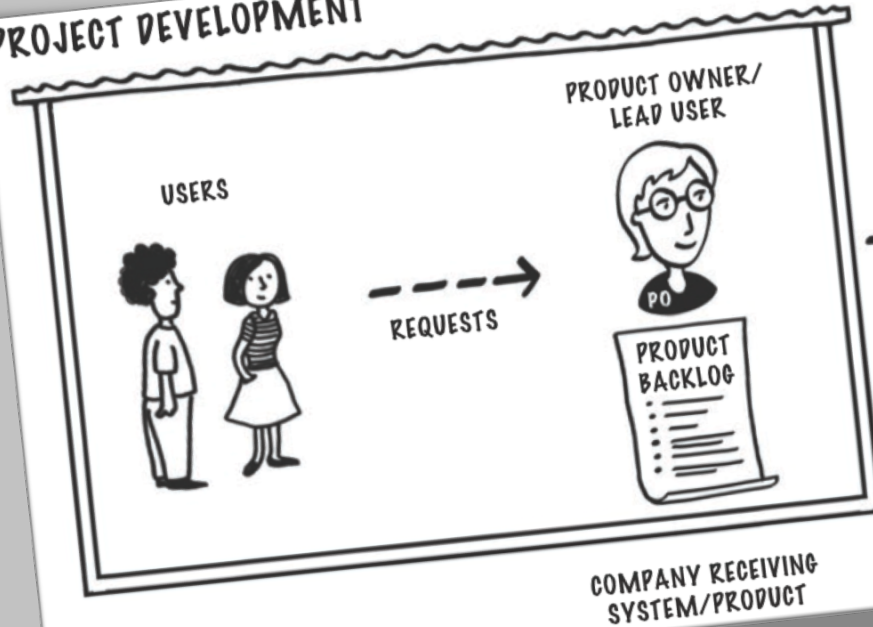


INTERNAL (PRODUCT) DEVELOPMENT



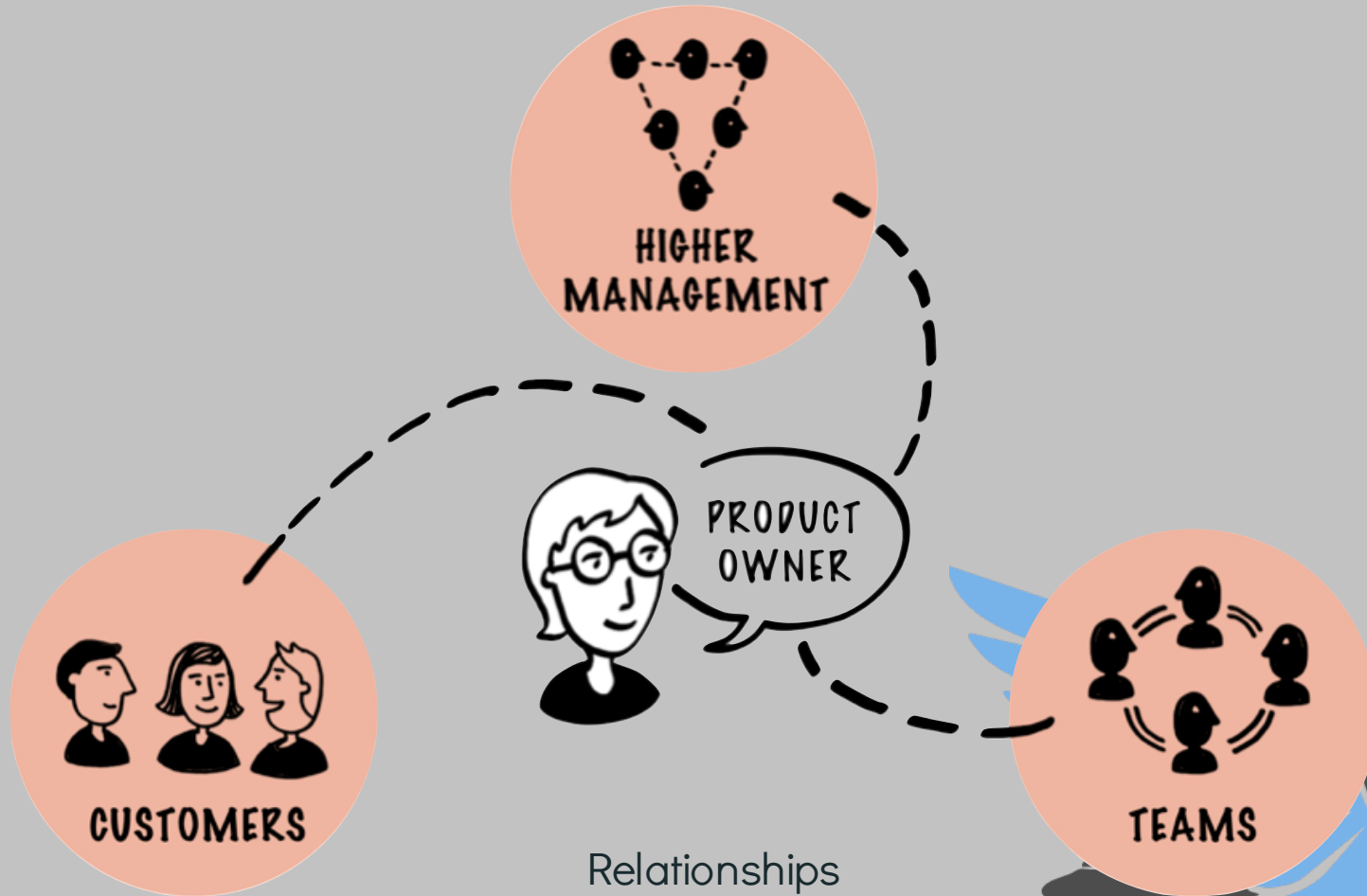
Internal Product Development

PROJECT DEVELOPMENT



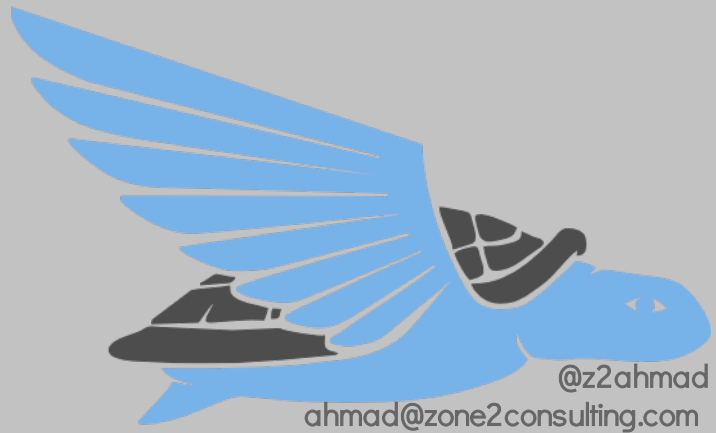
FEATURE
PRIORITY





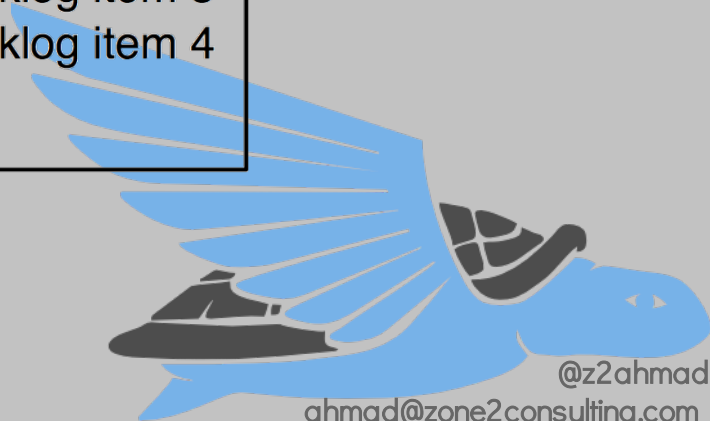
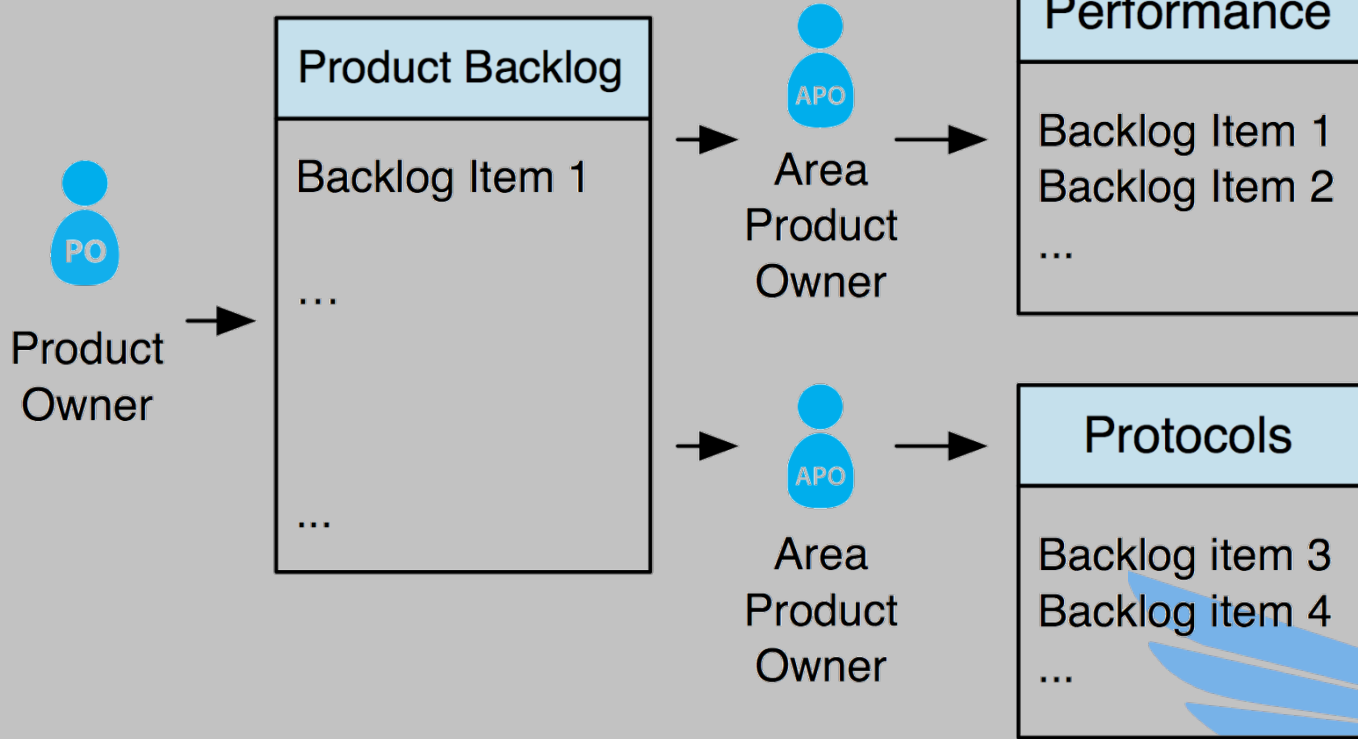
Temporary Fake PO

www.ahmadfahmy.com



@z2ahmad

ahmad@zone2consulting.com



Get help with:

- Administrative work
- Relationships to customers or senior management
- Competitive intelligence
- Visioning, future technology adoption

Prefer help from the teams

- Work together and add items to Product Backlog

Alternatively get help from Product Owner Team

- In LeSS: Product Owner and people who support him.
- In LeSS Huge: Also includes all Area Product Owners





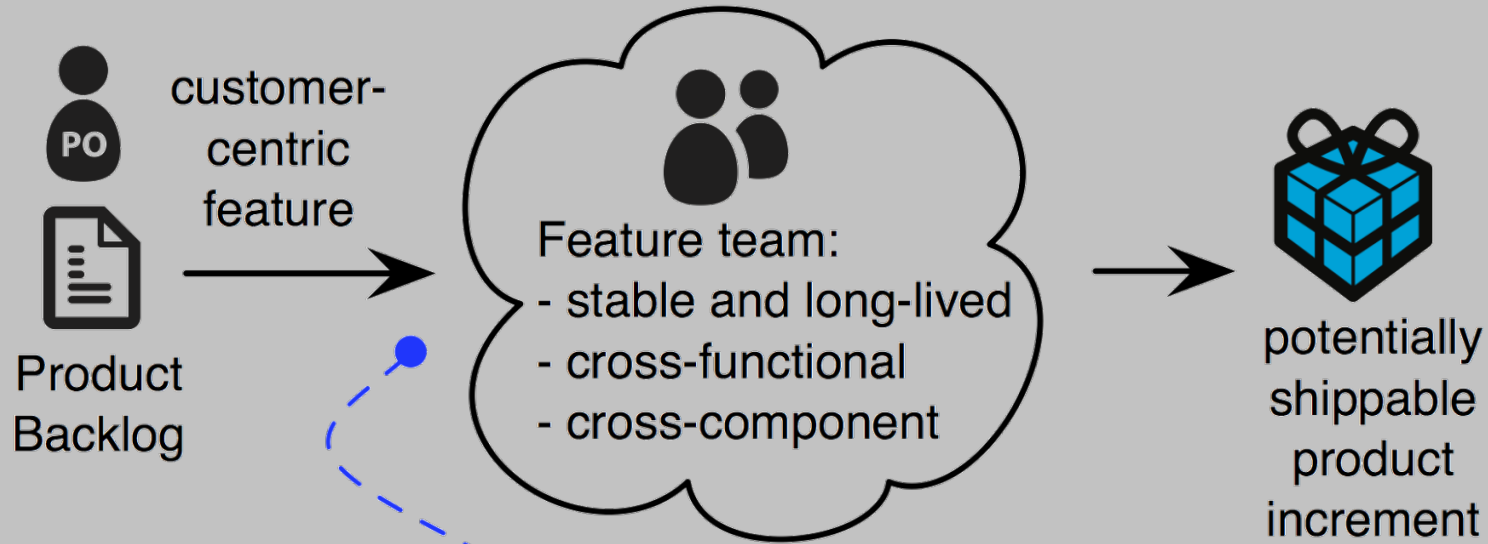
Area Product Owner

www.ahmadfahmy.com

A black and white photograph of the four members of the hip-hop group The Notorious B.I.G. The group consists of four men. On the left is a Black man with a beard and a shaved head, wearing a chain around his neck and a large bracelet. In the center back is a white man with a baseball cap, smiling. In the center front is a white man with short hair, smiling. On the right is a white man with short hair, looking towards the camera. The text "Feature team" is overlaid in a large, bold, black font across the middle of the image.

Feature team

Feature Team



Team has the necessary knowledge and skills to complete an end-to-end customer-centric feature. If not, the team is expected to learn or acquire the needed knowledge and skill.

component team	feature team
optimized for delivering the maximum number of lines of code	optimized for delivering the maximum customer value
focus on increased individual productivity by implementing 'easy' lower-value features	focus on high-value features and system productivity (value throughput)
responsible for only part of a customer-centric feature	responsible for complete customer-centric feature
traditional way of organizing teams — follows Conway's law	'modern' way of organizing teams — avoids Conway's law
leads to 'invented' work and a forever-growing organization	leads to customer focus, visibility, and smaller organizations
dependencies between teams leads to additional planning	minimizes dependencies between teams to increase flexibility
focus on single specialization	focus on multiple specializations
individual/team code ownership	shared product code ownership
clear individual responsibilities	shared team responsibilities
results in 'waterfall' development	supports iterative development
exploits existing expertise; lower level of learning new skills	exploits flexibility; continuous and broad learning
works with sloppy engineering practices—effects are localized	requires skilled engineering practices—effects are broadly visible
contrary to belief, often leads to low-quality code in component	provides a motivation to make code easy to maintain and test
seemingly easy to implement	seemingly difficult to implement

Component vs Feature team



The Local Optimization, Local Efficiency Thinking Mistake

“it’s more **efficient/productive** when a person/group does one specialization”



@less_works

Component vs Feature team

www.ahmadfahmy.com



@z2ahmad

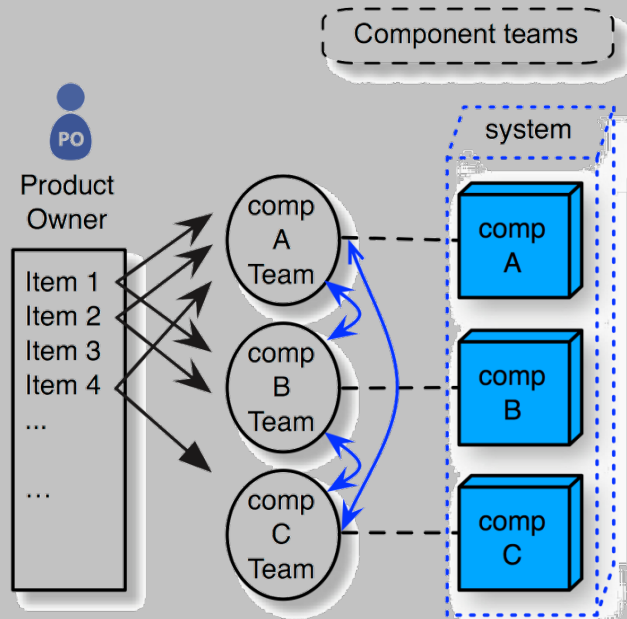
ahmad@zone2consulting.com



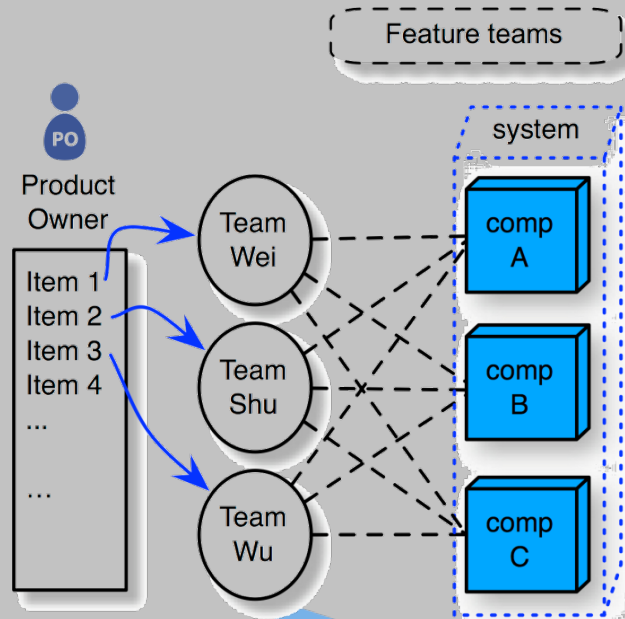
You are not your role

Draw your personal map



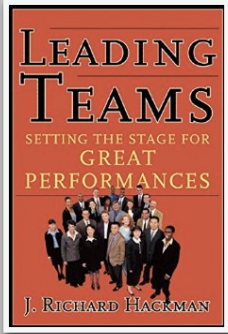


Work from multiple teams is required to finish a customer-centric feature. These dependencies cause waste such as additional planning and coordination work, hand-offs between teams, and delivery of low-value items. Work scope is narrow.



Every team completes customer-centric items. The dependencies between teams are related to shared code. This simplifies planning but causes a need for frequent integration, modern engineering practices, and additional learning. Work scope is broad.

Component vs Feature team



Setting overall direction

Designing the team and its organizational context

Monitoring & managing work process & progress

Executing the team task

Management Responsibility			
	Team's Own Responsibility		

Manager-led teams

Self-Managing teams

Self-Designing teams

Self-Governing teams

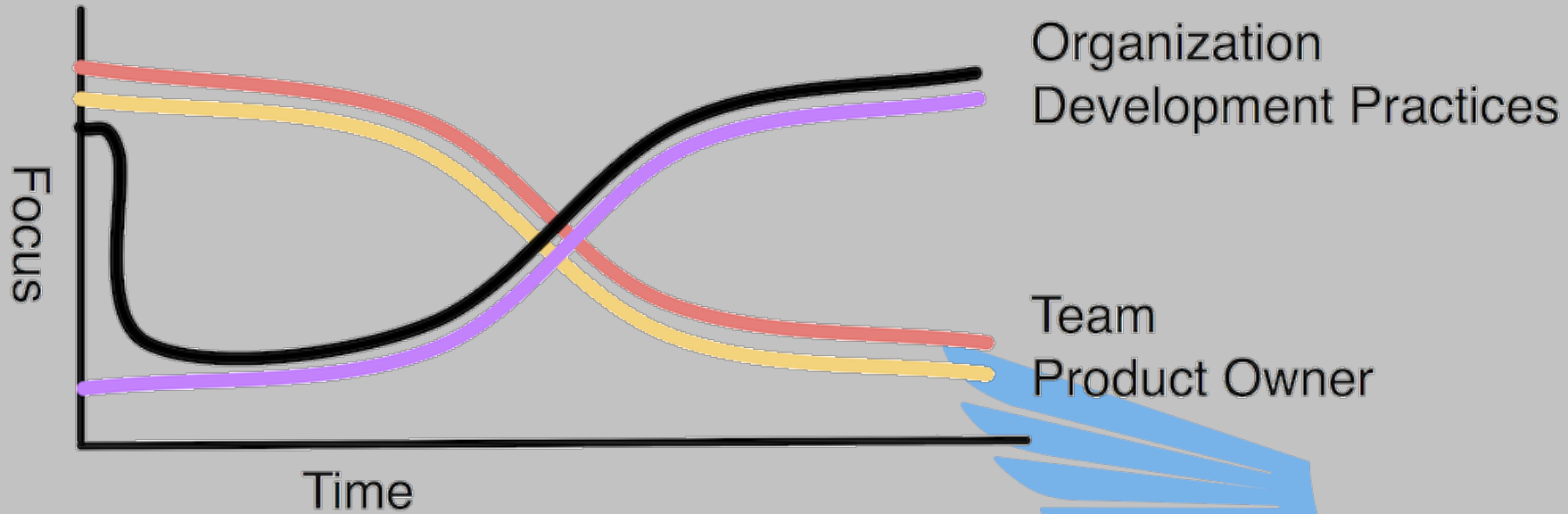




SCRUM MASTER

~~ScrumMaster
represents
a team at
meetings (SoS,)~~





Time Focus over time



customer

real external paying
customers

or hands-on users



@less_works

www.ahmadfahmy.com



@z2ahmad

ahmad@zone2consulting.com



MANAGEMENT

BB

**“There is a no question that cost of production is lowered by separating the work of planning and the brain work as much as possible from manual labour”
- Fredrick Taylor**

"We will win and you will lose. You cannot do anything about it because your failure is an internal disease. Your companies are based on Taylor's principles. Worse, your heads are Taylorized, too. You firmly believe that sound management means executives on one side and workers on the other, on one side men who think and on the other side men who can only work. For you, management is the art of smoothly transferring the executives' ideas to the workers' hands."

Konosuke Matsushita (1)



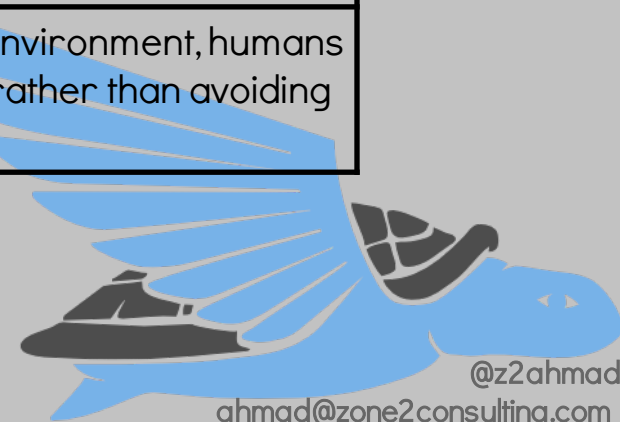
“We have passed the Taylor stage. We are aware that business has become terribly complex. Survival is very uncertain. Therefore, a company must have the constant commitment of the minds of all of its employees to survive. For us, management is the entire workforce's intellectual commitment at the service of the company.

We know that the intelligence of a few technocrats—even very bright ones—has become totally inadequate to face these challenges. Only the intellects of all employees can permit a company to live with the ups and downs and the requirements of its new environment. Yes, we will win and you will lose. For you are not able to rid your minds of the obsolete Taylorisms that we never had.”

Konosuke Matsushita (2)

Theory X	Theory Y
Average humans dislike work and will try to avoid working.	People spend effort to work as natural as they do to play and rest.
Because of this, people will need to be coerced, controlled, directed and threatened so that the maximum amount of effort can be extracted out of them.	People will use self-direction and self-control for goals that he is committed to. Commitment comes most strongly from the intrinsic rewards related to the achievement itself. That is the challenge, the learning, and the sense of purpose.
People want to be directed as they have little ambition and avoid taking responsibility.	Provided the right environment, humans seek responsibility rather than avoiding it.

Theory X vs Theory Y

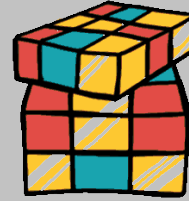




ROLE OF MANAGER



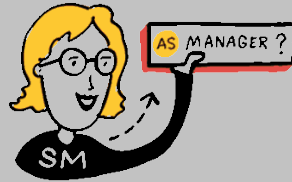
GO SEE



TEACHING PROBLEM
SOLVING



MANAGEMENT

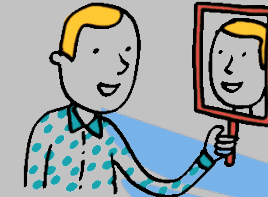


MANAGER
AS SCRUMMASTER?

IMPROVEMENT
SERVICE



Management



SELF-MANAGEMENT



manager-teacher



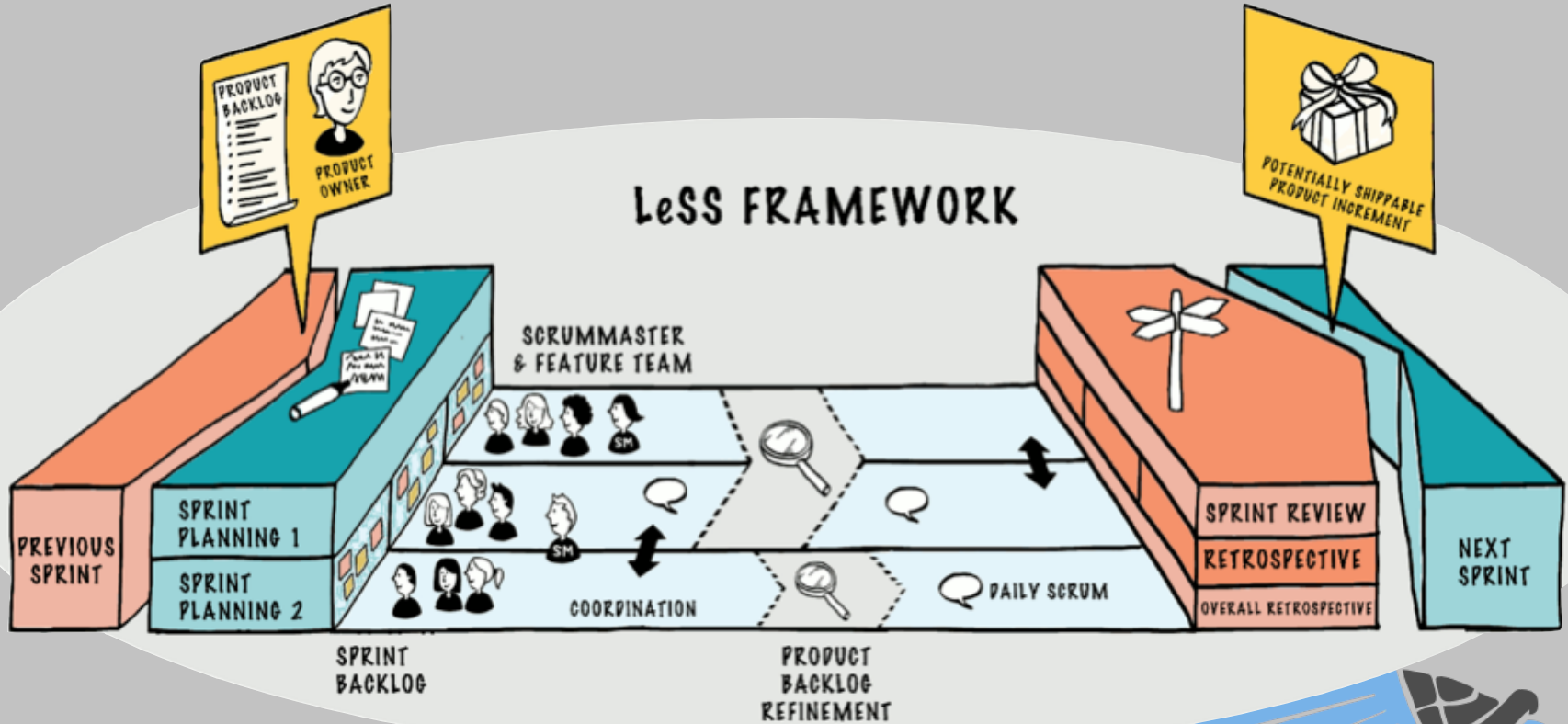
“my manager can do my
job better than me”
Toyota (Lean) saying

“One of the real problems
with American Business is
this notion that you can be
trained in management, in
some kind of generic form
of management, and that
then you can manage any
operation.

But that absolutely
doesn't work in a
technical situation.”

Robert Noyce -
Founder of Intel

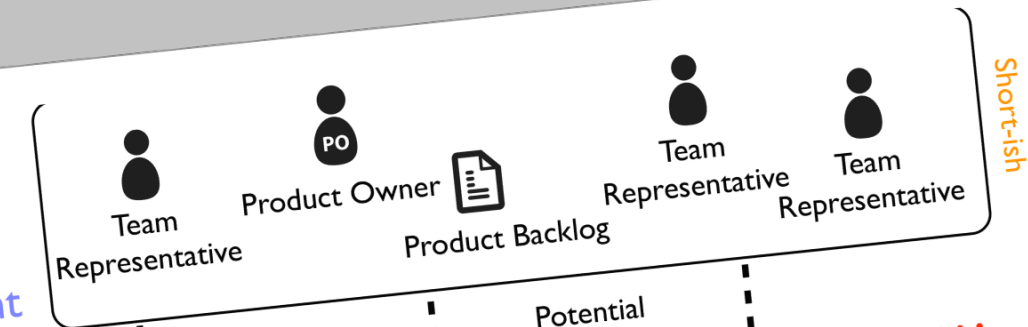
LeSS FRAMEWORK





Initial Product Backlog refinement

Overall
Product
Backlog
Refinement



Short-ish

Product
Backlog
Refinement

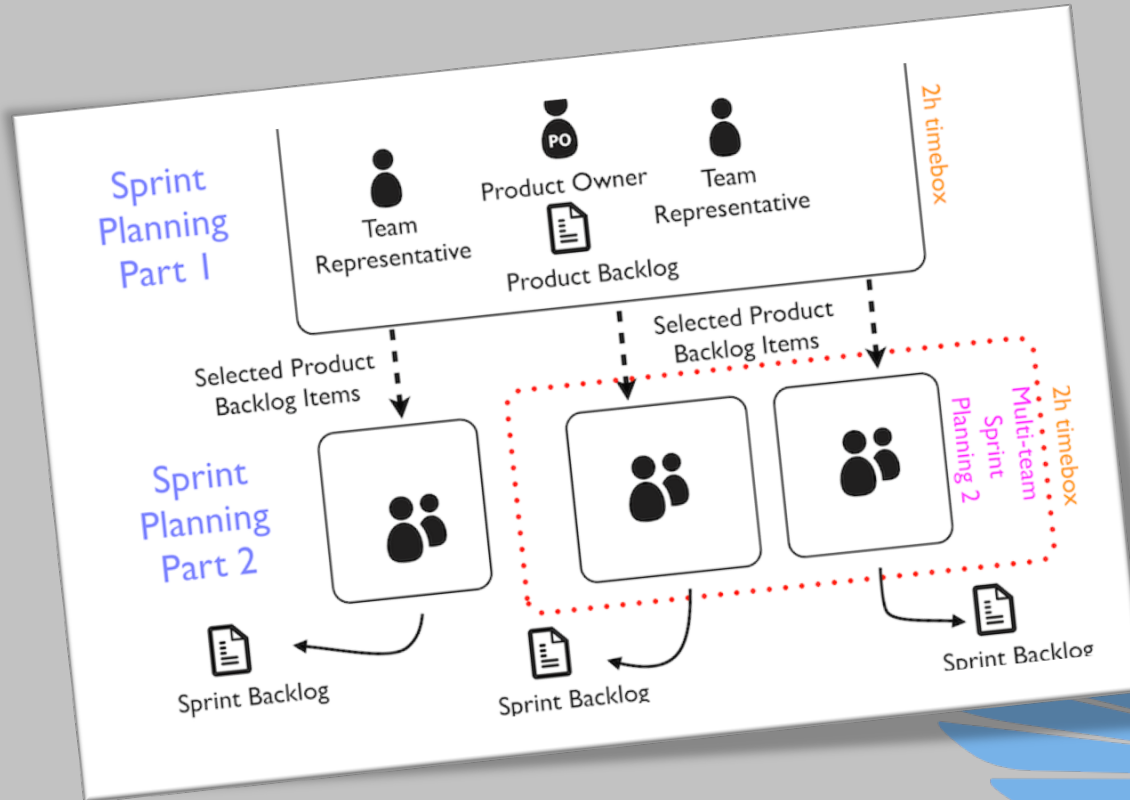
Potential
Items

Potential
Items

5-10% Sprint
Multi-team
Backlog
Refinement

Product Backlog

Product Backlog Refinement





Daily Scrum



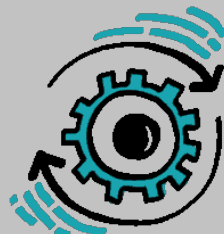
SPECIFICATION BY EXAMPLE



CONTINUOUS
INTEGRATION



CONTINUOUS DELIVERY



TEST AUTOMATION



TECHNICAL
EXCELLENCE



ARCHITECTURE
& DESIGN



ACCEPTANCE
TESTING



THINKING ABOUT TESTING



TEST-DRIVEN DEVELOPMENT

CODE

CLEAN CODE



UNIT TESTING

CC BY-NC-ND



THREE PRINCIPLES



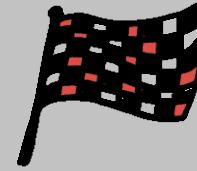
COACHING



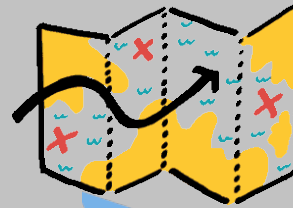
ADOPTION



CONTINUOUS IMPROVEMENT



GETTING STARTED



FEATURE TEAM
ADOPTION MAP



Adoption



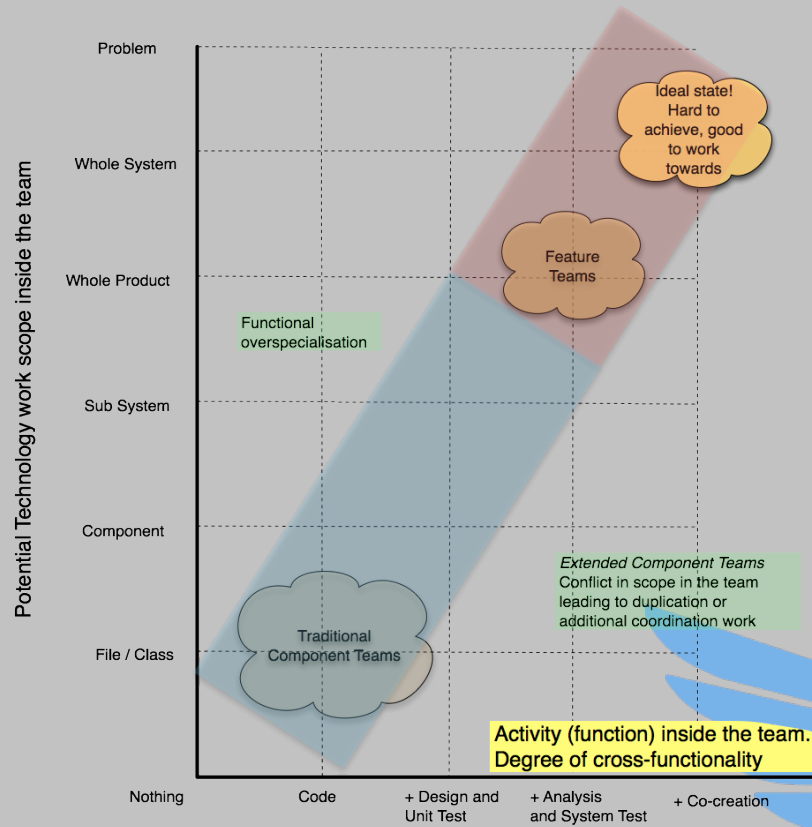
@less_works

www.ahmadfahmy.com

@z2ahmad

ahmad@zone2consulting.com



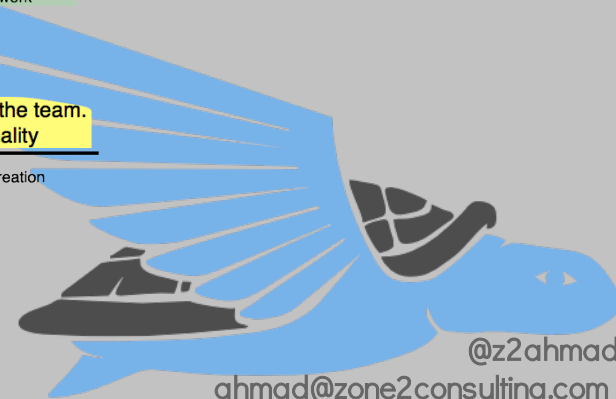


Feature adoption Map

www.ahmadfahmy.com

LeSS

@less_works

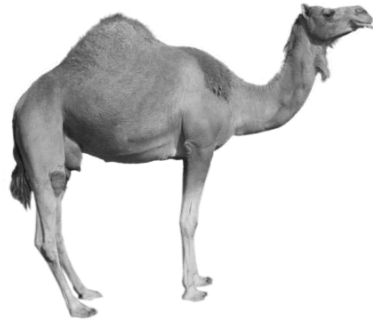


@z2ahmad

ahmad@zone2consulting.com

	0 PPP			Happy Business			Happy team			10% time improving product			Business drives priority			Production every two weeks			0 red metrics		
	Sense of ownership. Quality, craftsmanship			Delivery. Transparency.			Sustainable work hours. Constant learning. Working on interesting work.			Improving the product			Break the contract game.			More frequent delivery of value					
	Production support			The business			The team			The business			Business owns the list			Production support			Business management		
	JULY	AUG	2015	JULY	AUG	2015	JULY	AUG	2015	JULY	AUG	2015	JULY	AUG	2015	JULY	AUG	2015	JULY	AUG	2015
Product 1	5	2	0	:)	:)	:)	:)	:(:)	0%	0%	0%	no	no	yes	3	2	2	0	0	0
product 2																					
Product 3																					
Product 4																					
Product 5																					
Product 6																					
Product 7																					
Product 8																					
Product 9																					
Product 10																					

Metrics

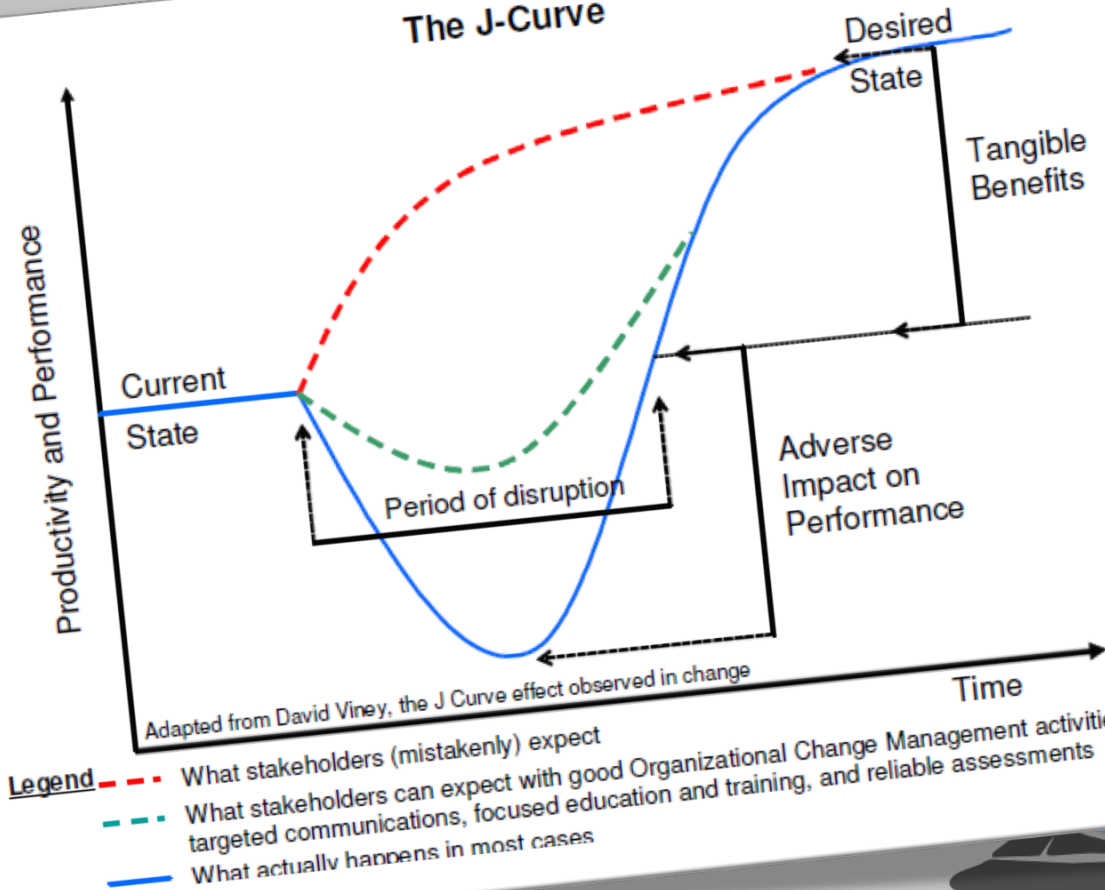


Adoption

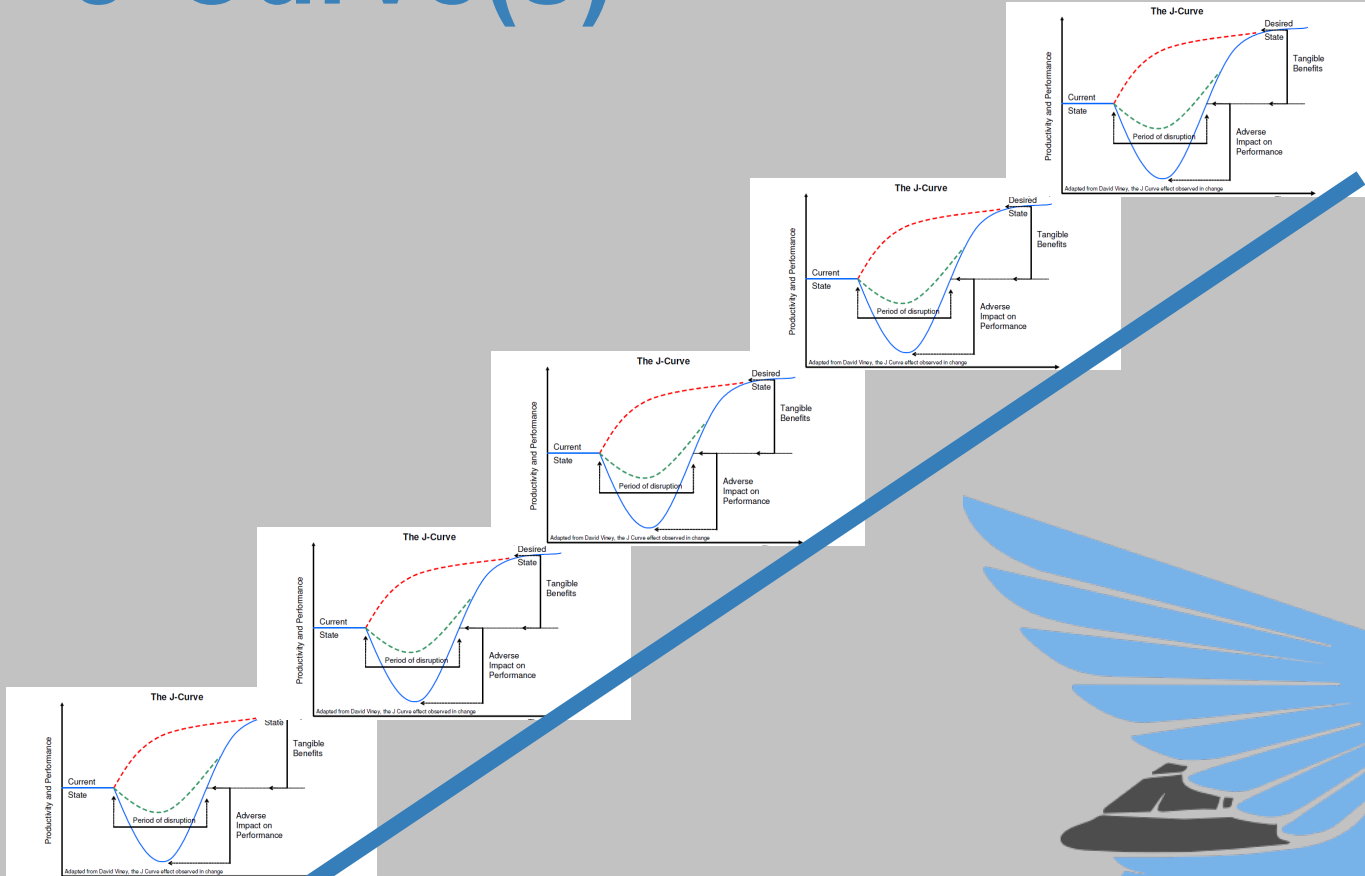
Transformation

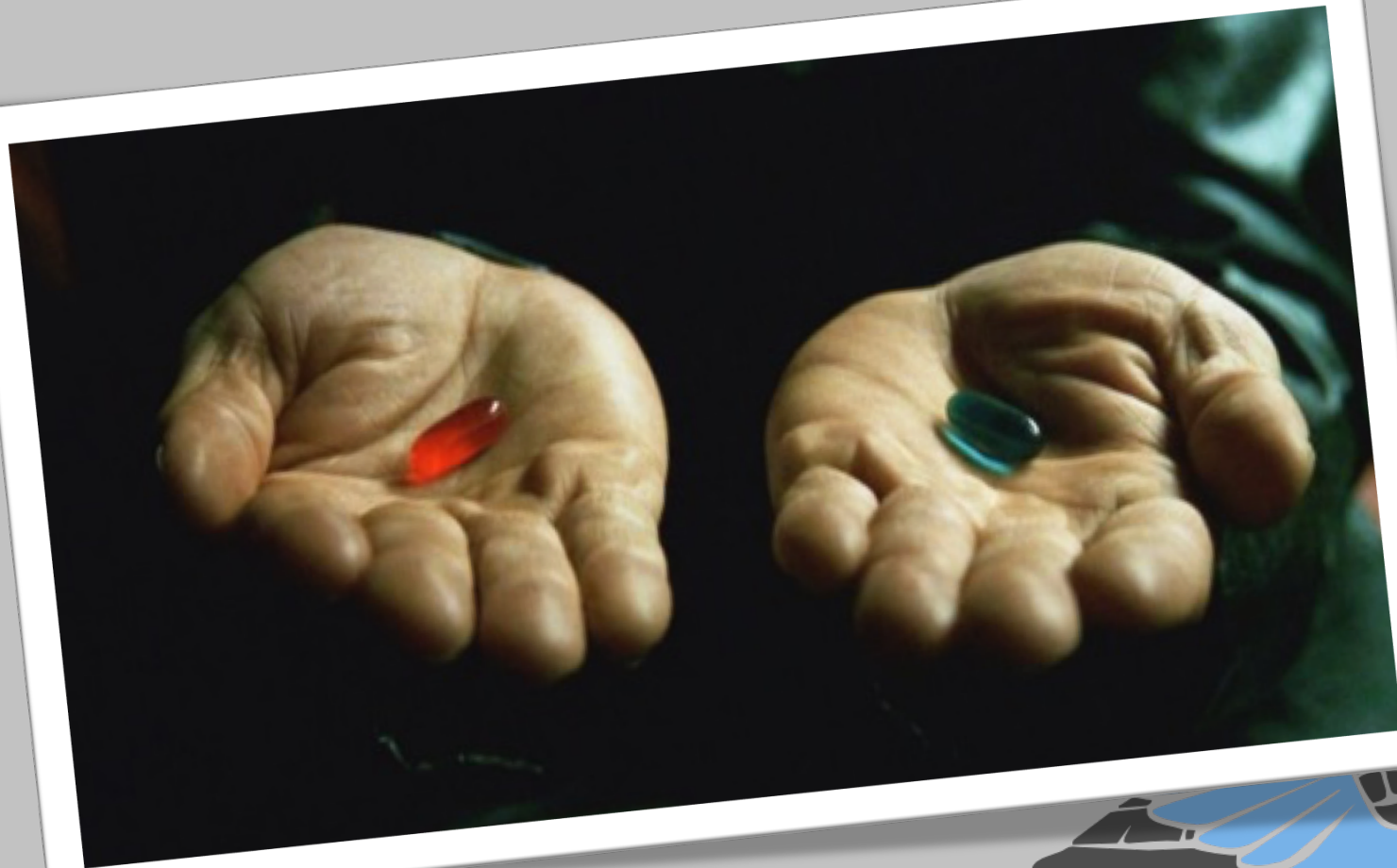


The J-Curve

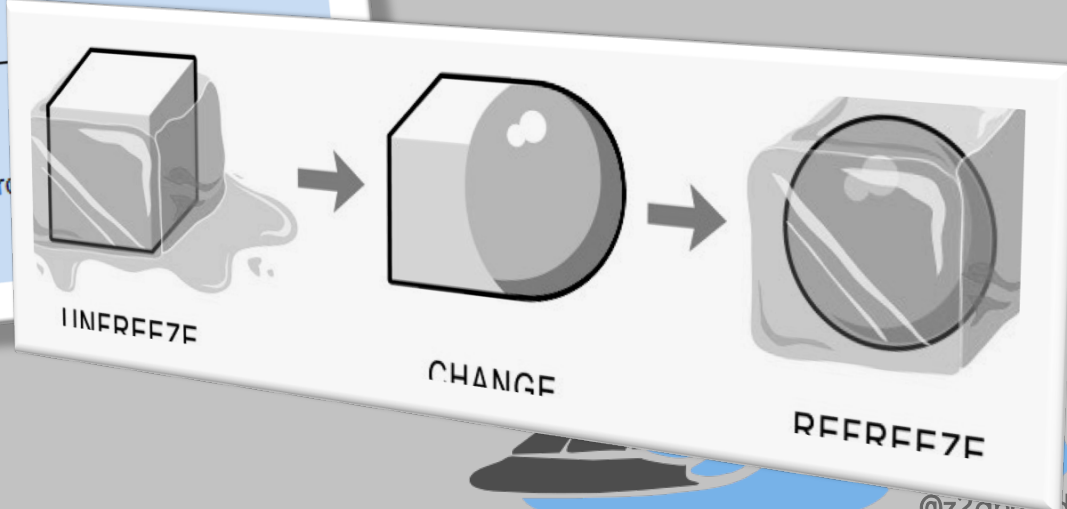


J-Curve(s)





Kurt Lewin	John Kotter
Unfreeze	<ol style="list-style-type: none"> 1. Establish a Sense of Urgency 2. Create the Guiding Coalition 3. Develop a Vision and Strategy 4. Communicate the Change Vision
Change	<ol style="list-style-type: none"> 5. Empower Broad-Based Action 6. Generate Short Term Wins 7. Consolidate Gains & Make More Change
Refreeze	<ol style="list-style-type: none"> 8. Anchor New Approaches



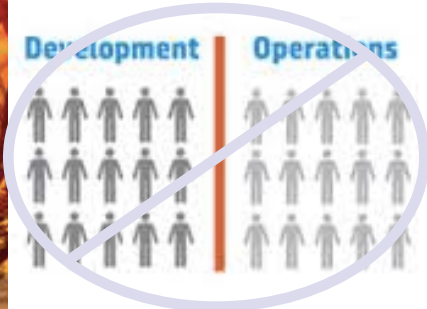
- steering adaptively by true business-side Product Owner, not a Development-side project manager
- there is no contract/commitment with Development to “deliver the project scope”
- the end of “projects” & “programs” in Development
 - -> product-centric, not project-centric
- no control by project manager; project mgmt. responsibilities taken over by teams & business-side PO
- no “status reporting by the project managers”
- no big batch or long release
 - release every Sprint (or as close as possible)



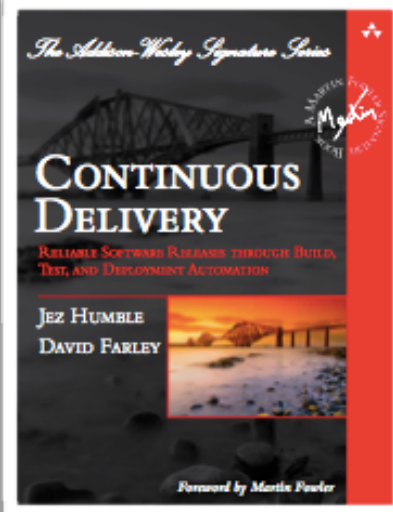


you should NOT need a
Release Sprint or
Undone Department;
but these may be a
temporary “necessary evil”
during early transition to
LeSS

CONTINUOUS DELIVERY



BLOG PUBLICATIONS TALKS ABOUT



« [On Antifragility in Systems and Organizational Architecture](#)

[Elisabeth Hendrickson Dis](#)

There's No Such Thing as a "Devops Team"

Nor should there be ["devops specialists"](#)

Why Segregation of Duties Doesn't Work

Why are Functional Silos Problematic?

“change management”

“change manager”

“change team”

“change project”



Miscellaneous Preparations

- ▮ robust & fast build system with strong feedback mechanisms

- ▮ version control capable of supporting **integrating continuously**

- ▮ physical environment: common table for each team, whiteboards, ...

- ▮ moving people together

- ▮ identification and preparation around coaching, courses



@less_works

www.ahmadfahmy.com

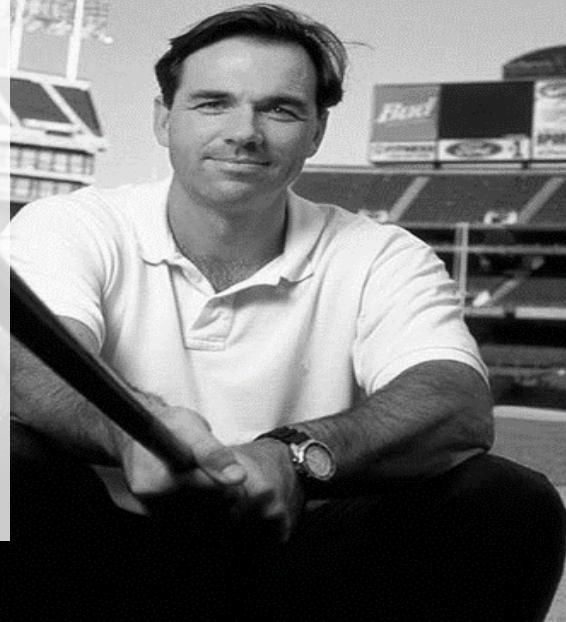


@z2ahmad

ahmad@zone2consulting.com

Lesson 3 Get *real* wins early

- **Doing Scrum/Kanban are not wins. Shipping software is. Don't declare success too early**
- **Billy Bean introduced a new way of selecting players and was shunned by the baseball establishment**
- **20 game win streak despite having one of the lowest pay rolls in the industry**
- **Changed the way baseball is played**



Lesson 4 Hire the best

- Process is not enough
- Hire the best engineers
- Hire great product people
- Hire great coaches



Congratulations!

- Access to the less.works book
- Draft of the new “large-scale Scrum” book
- Training material
- Stay in contact with class members
- Please give feedback and tell us what you would want!



I will register you at less.works

what you will find there



Connections

- LeSS Site:
<http://less.works>
- LinkedIn Group: Certified LeSS Practitioner
- LeSS Discussion Group:
<https://groups.google.com/forum/#!forum/largescalescrum>
- LeSS Twitter:
#LeSSWorks, @less_works



@less_works

www.ahmadfahmy.com



@z2ahmad

ahmad@zone2consulting.com

take the posters home



@less_works

www.ahmadfahmy.com



@z2ahmad

ahmad@zone2consulting.com

group photo



@less_works

19

www.ahmadfahmy.com



@z2ahmad

ahmad@zone2consulting.com

team/class do:
final feedback review



@less_works

www.ahmadfahmy.com



@z2ahmad

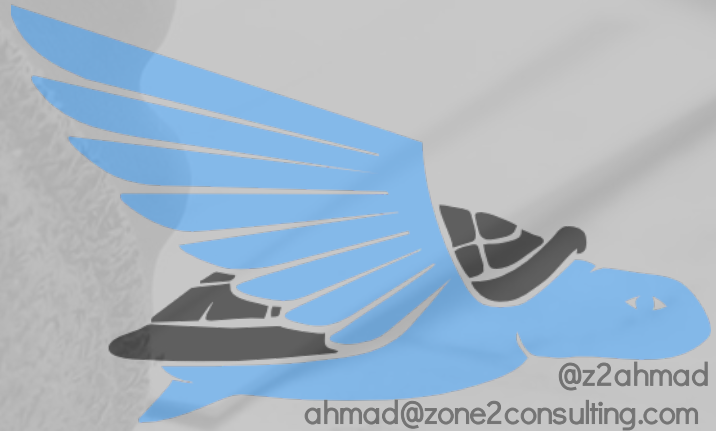
ahmad@zone2consulting.com

THANKS!

You can find me at @z2ahmad & ahmad@zone2consulting.com


@less_works

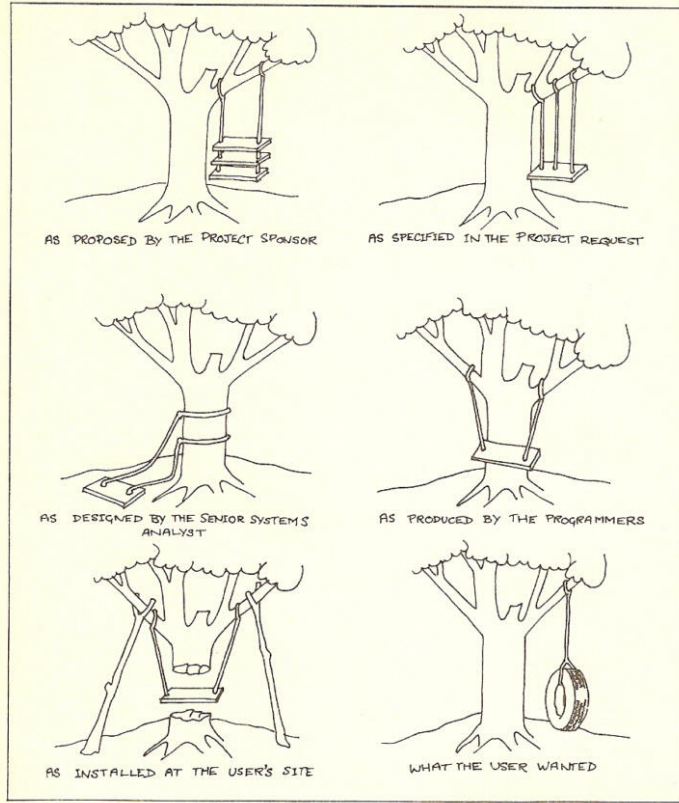
www.ahmadfahmy.com



@z2ahmad
ahmad@zone2consulting.com

1. Organizations are implicitly optimized to avoid changing the status quo middle- and first-level manager and “specialist” positions & power structures.
2. As a corollary to (1), any change initiative will be reduced to overloading or redefining the new terminology to mean basically the same as status quo.
3. As a corollary to (1), any change initiative will be derided as “purist”, “theoretical”, and “needing pragmatic customization for local concerns” — which deflects from addressing weaknesses and manager/specialist status quo.
4. Culture follows structure (or culture follows system)





From the *University of London Computer Centre Newsletter* No. 53, March 1973

most of you have seen this cartoon

notice the **date**

what does this say about what was
widely understood even by that time?

have you seen this mistake **even**
today?



@z2ahmad

ahmad@zone2consulting.com