

Agile Overview

References

Book & Edition	Author	Publications
PMI-ACP® Exam Prep, 2012	Mike Griffiths	RMC
Coaching Agile Teams, 2010	Lyssa Adkins	Addison Wesley
The Art of Agile development, 2007	James Shore	O'Reilly

Online	Content
http://guide.agilealliance.org/ http://agiledictionary.com/	Agile Dictionary
https://www.scrum.org/Scrum-Guide	Authentic Scrum Guide
http://www.jamesshore.com/Agile-Book/	“The Art of Agile” book By James Shore
http://en.wikipedia.org/wiki/Agile_software_development	Wikipedia
And many more on the net	

Common short-forms used

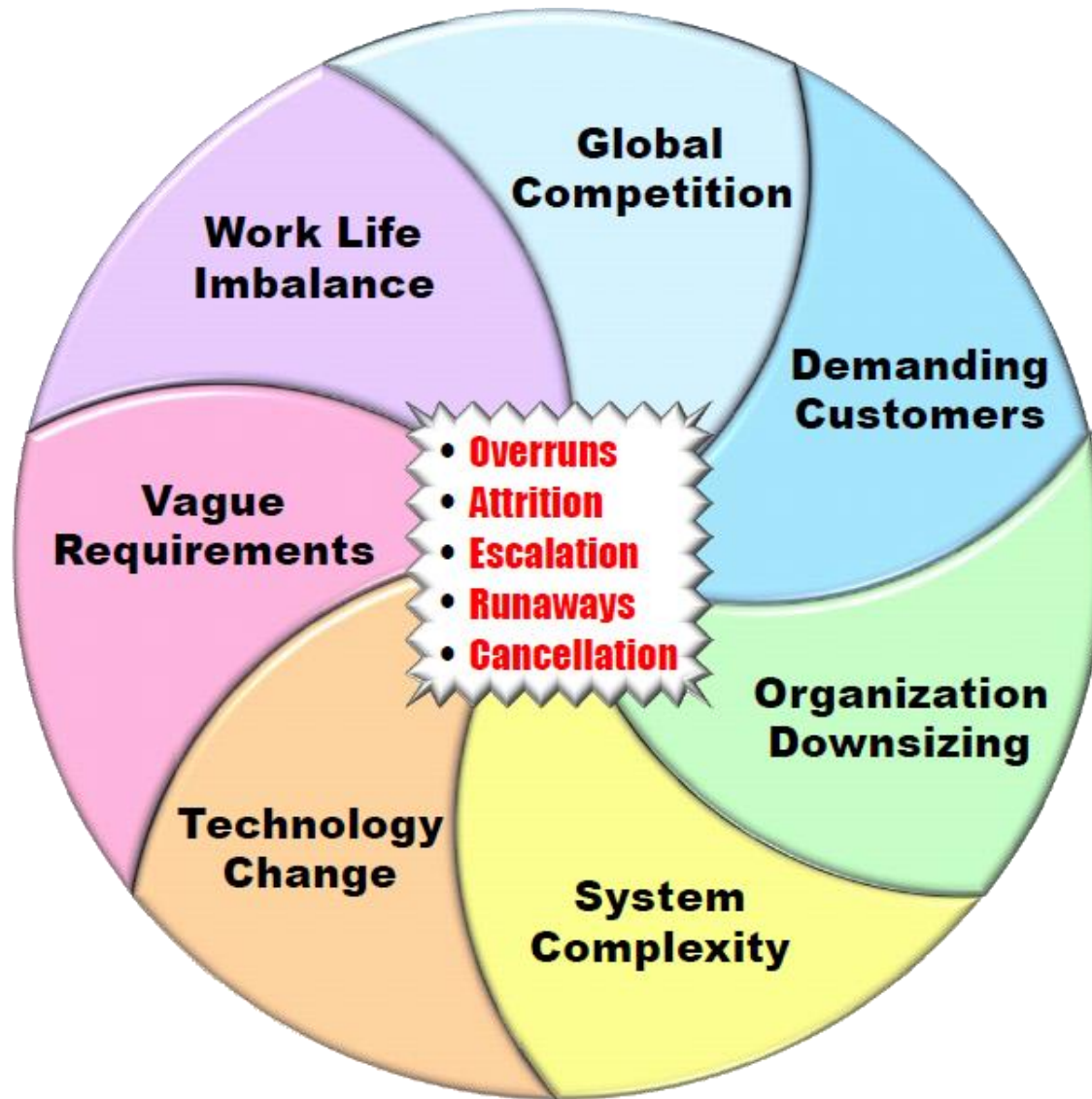
Short form	Full form	Short form	Full form
USP	Unique Selling Point	NPV	Net Present Value
TTM	Time To Market	IRR	Internal Rate of Return
SLA	Service Level Agreement	ROI	Return on Investment
PERT	Program Evaluation & Review Technique	T&M	Time & Materials (a form of contract)
JIT	Just In Time	FP	Fixed Price (a form of contract)
EVM	Earned Value Management	SOW	Statement of Work
SME	Subject Matter Expert	F2F	Face to Face
SWOT	Strength, Weakness, Opportunities, Threat	WIP	Work In Progress
Comm.	Communication	EI	Emotional Intelligence
Mgmt.	Management	Dev.	Development

Software / Agile short-forms used

Short form	Full form	Short form	Full form
SW	Software*	XP	Extreme Programming
DB	Database	FDD	Feature Driven Development
UI	User Interface	DSDM	Dynamic Systems Development Method
GUI	Graphical User Interface	SM	Scrum Master
SH	Stakeholder	PO	Product Owner
QA	Quality Assurance	PB	Product backlog
QC	Quality Control	MMF	Minimum Marketable Feature
CI	Continuous Integration	VSM	Value Stream Mapping
UT	Unit Testing	TDD	Test Driven Development
IT	Integration Testing	TFD	Test First Development
AT	Acceptance Testing	ATDD	Acceptance TDD
PMO	Project Management Office	* Can also mean Knowledge Worker Project in agile context	

2. AN OVERVIEW OF AGILE

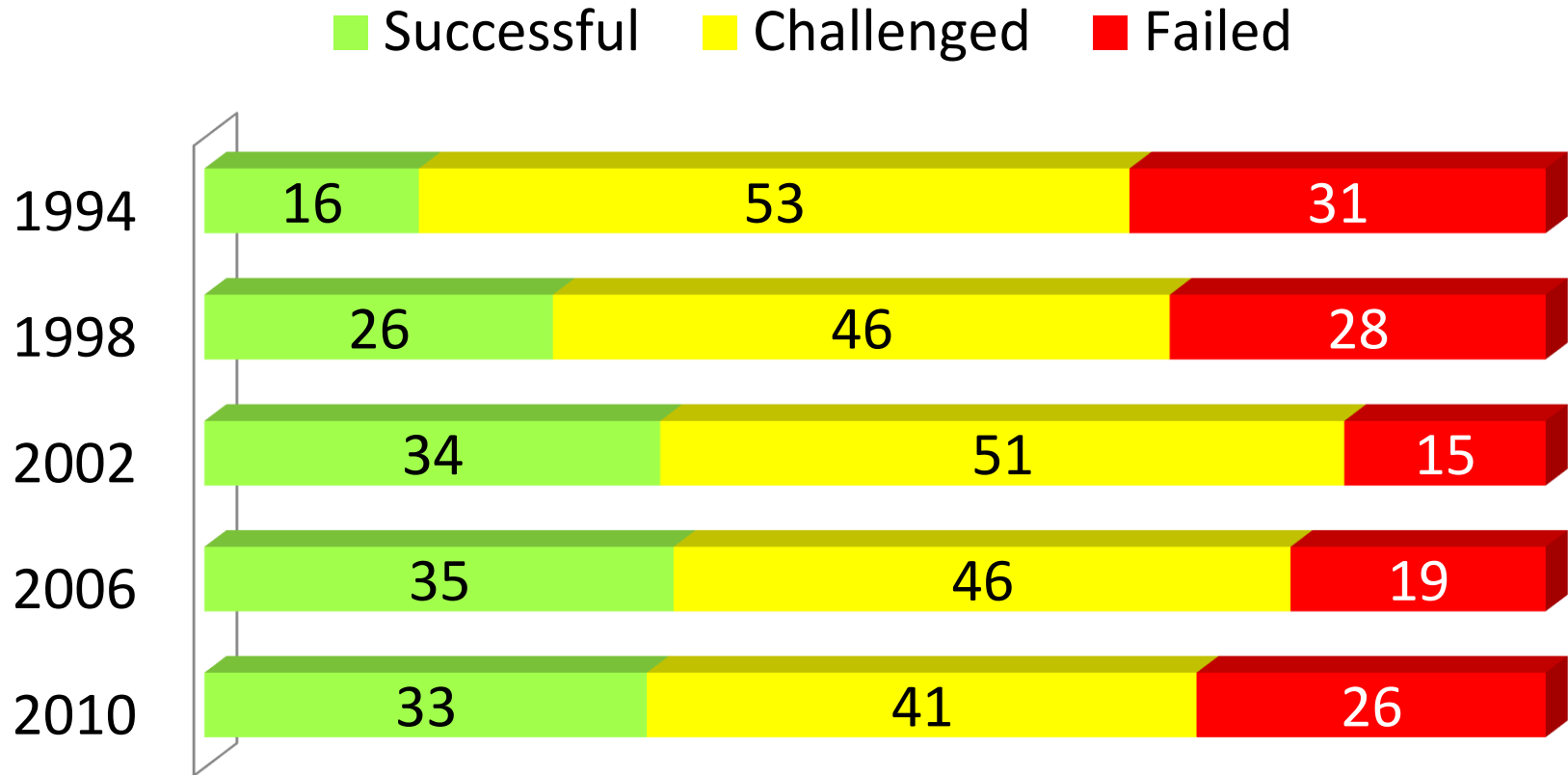
Today's whirlwind environment



**20-50%
Requirements
Volatility**

Global IT Project failures

- Failed and Challenged projects hover at 67%



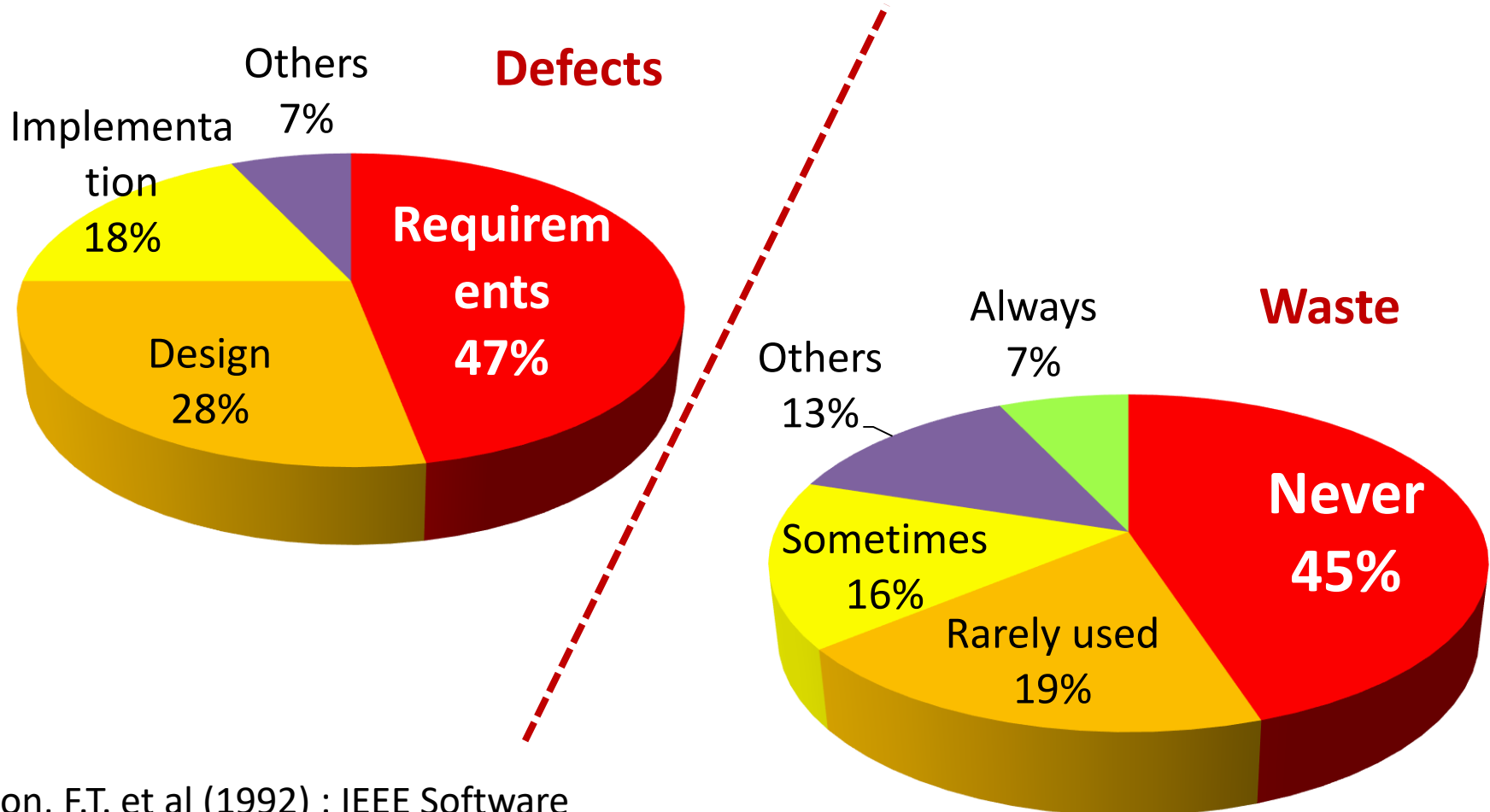
Failed: Cancelled OR Never used

Challenged: Delivery did not meet Time / Cost / Scope

Standish Group: Chaos manifesto, 2010

Requirements defects & waste

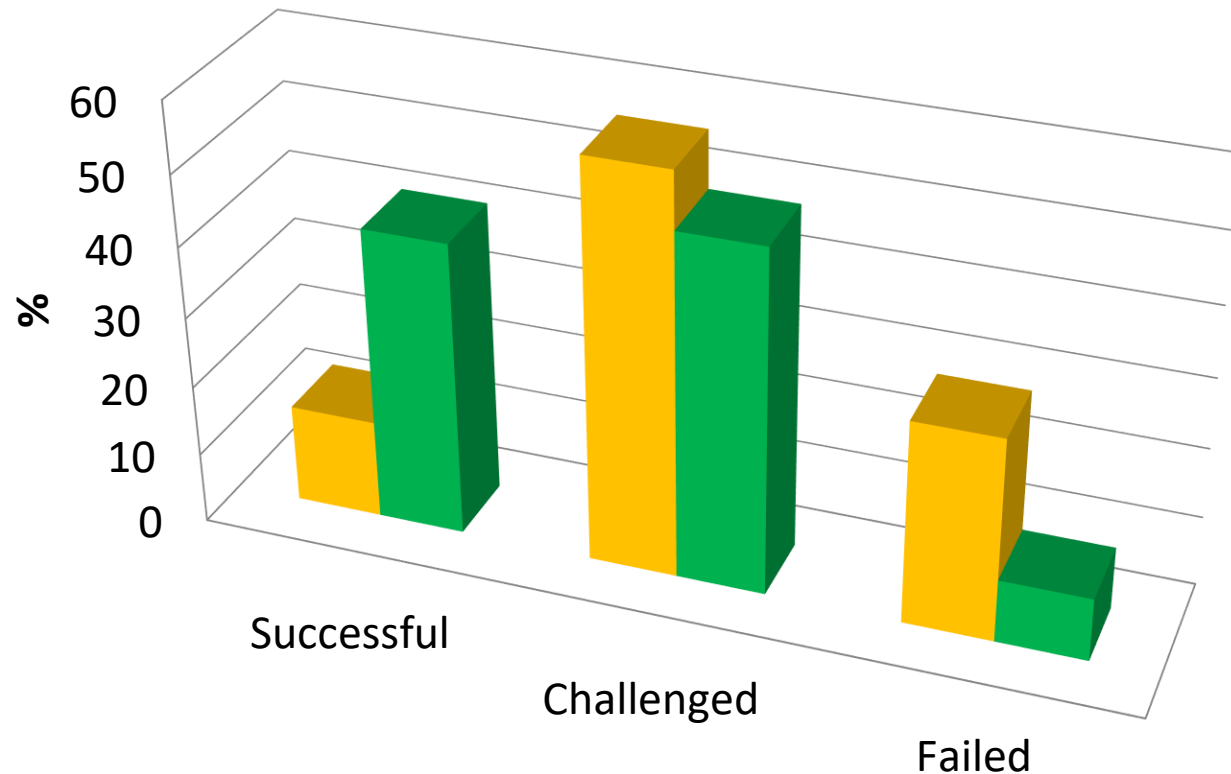
- Requirements defects are #1 reason for projects failure
- We build huge UNUSED features !!



Sheldon, F.T. et al (1992) : IEEE Software
Johnson J (2002). ROI: XP

Agile Vs. Waterfall

2011 CHAOS MANIFESTO: Results of **Waterfall Vs. Agile** from the CHAOS project database between 2002 to 2010



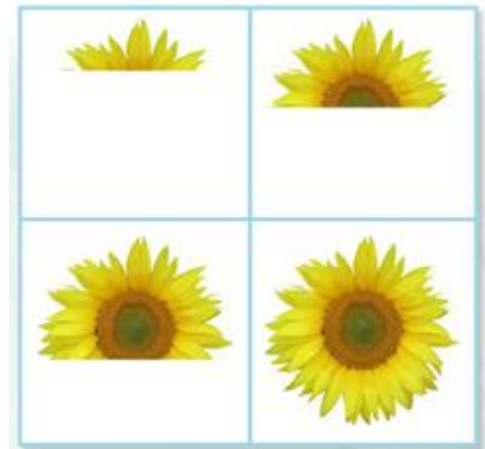
	Successful	Challenged	Failed
Waterfall	14	57	29
Agile	42	49	9

Agile Overview

1. Collection of **Light weight** Software development methods
 - XP, Scrum, DSDM, Crystal, FDD...
 - Aimed at reducing the risk of project failure by **Iterative & Incremental** development (early & frequent deliveries) and taking feedback from customer to ensure progress in right path
2. Continuous Customer Collaboration
3. Focus on VALUE to customer
4. **Rapid response to Change**

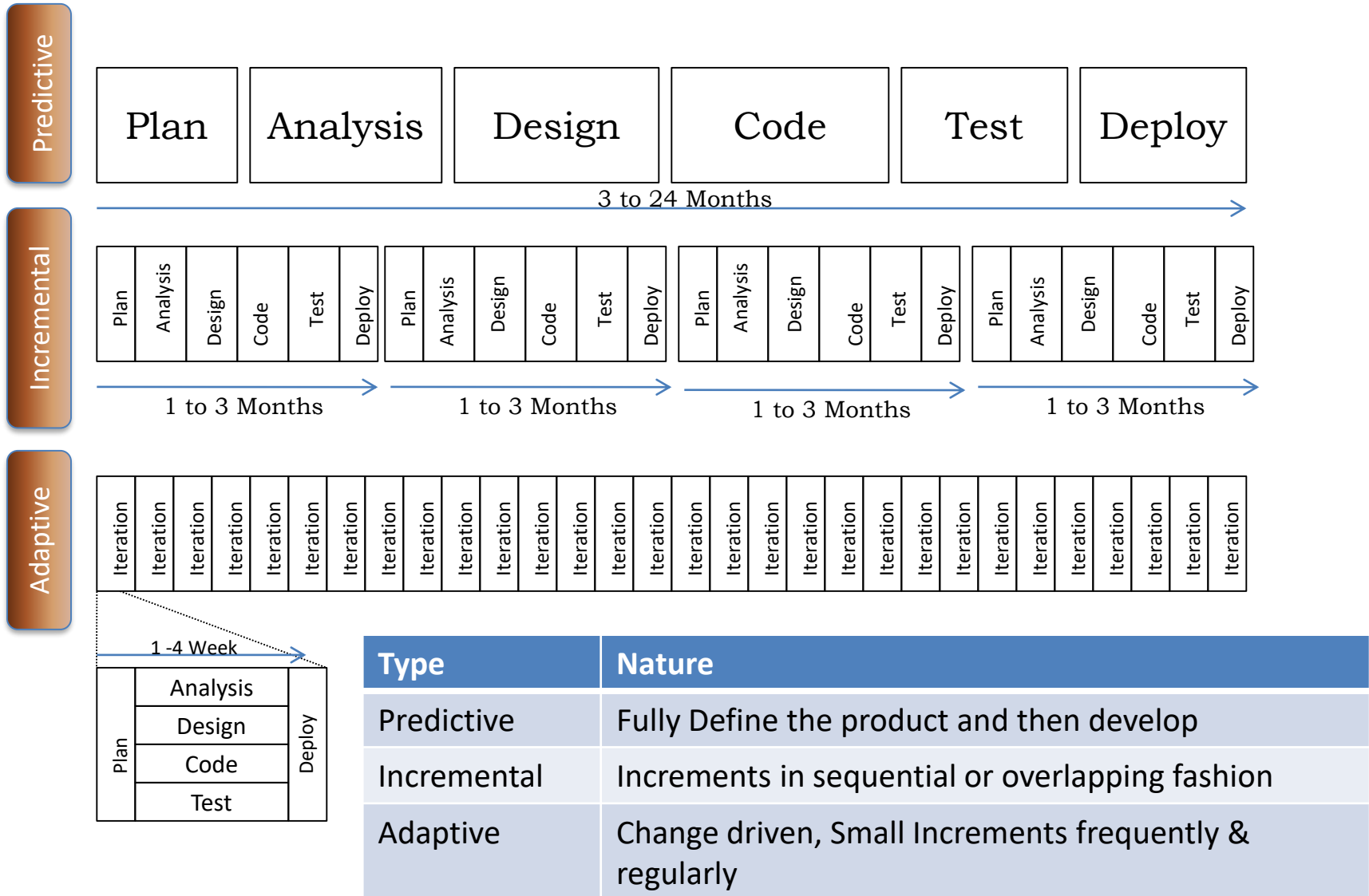


Iterative

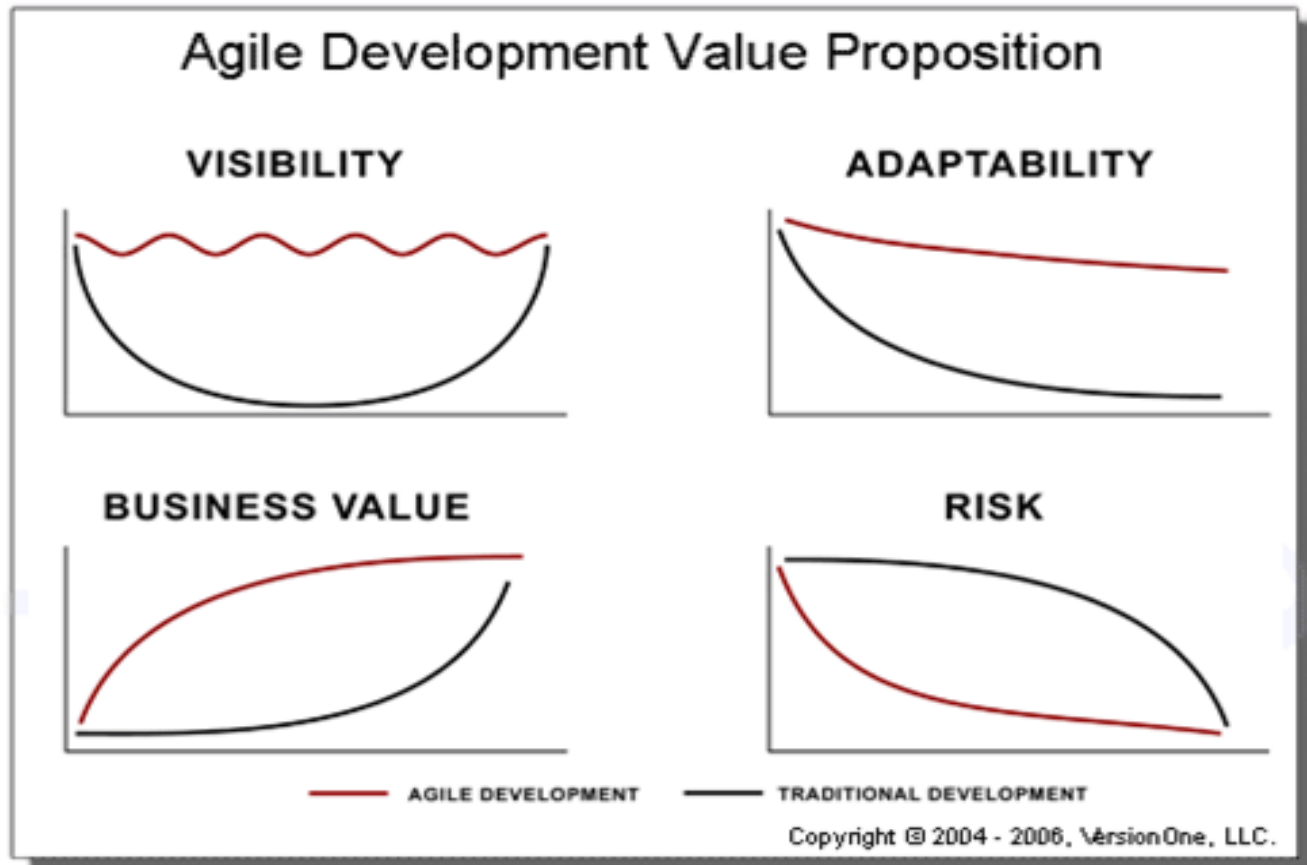


Incremental

Types of SW Project Life Cycle



Agile Vs Waterfall



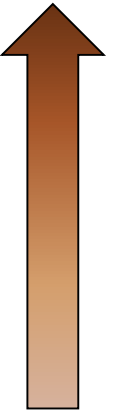
- Adaptive planning Vs Predictive planning
- Value driven Vs Process driven

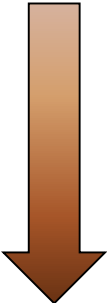
Agile benefits



Software Projects

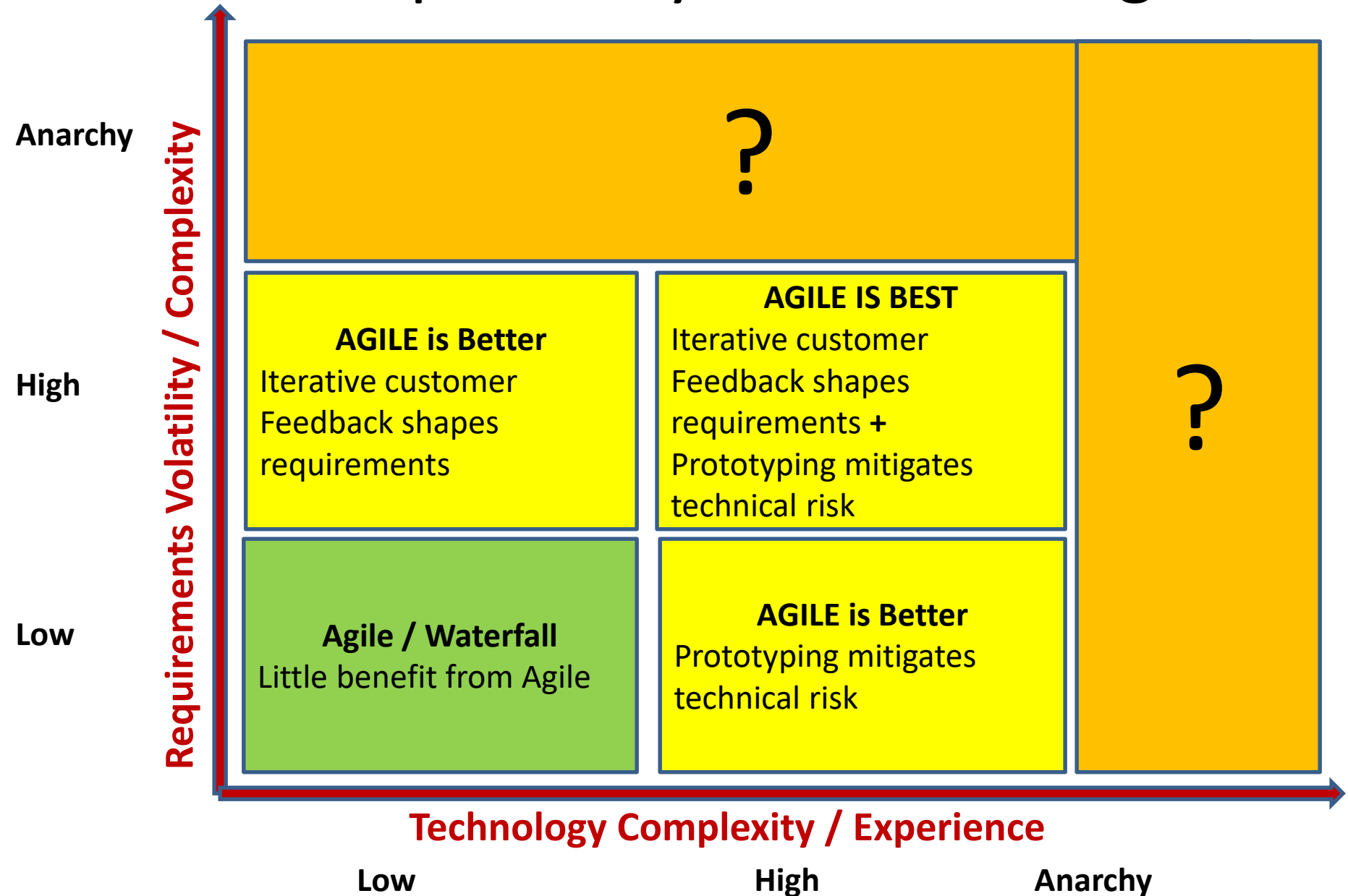
- ROI
- Success rate*
- Team motivation



- 
- A vertical arrow pointing downwards, colored with a gradient from light brown at the top to dark brown at the bottom.
- Time to Market
 - Risk
 - Defects

* 2011 CHAOS Manifesto from the Standish Group

Principles of systems thinking

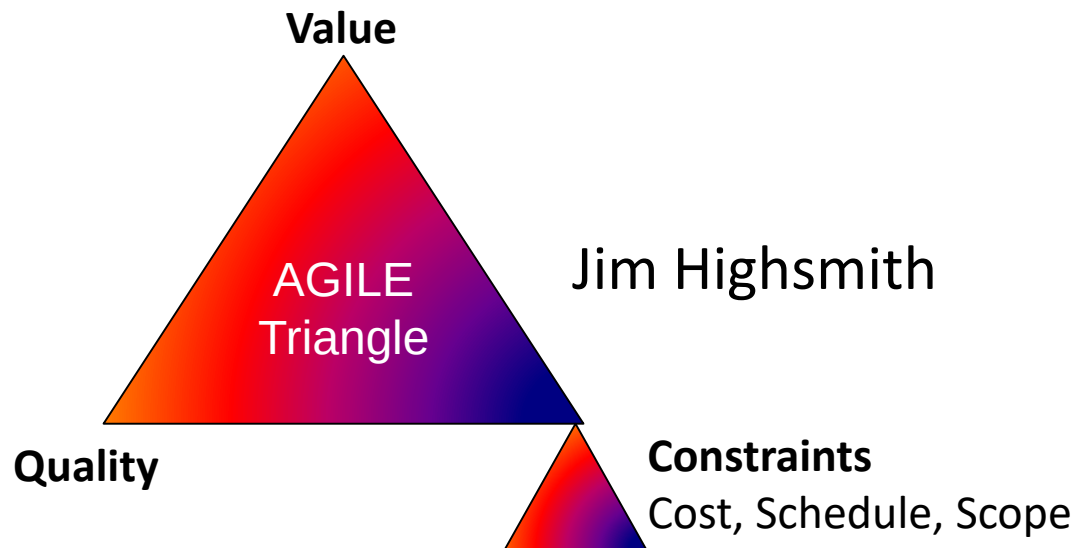
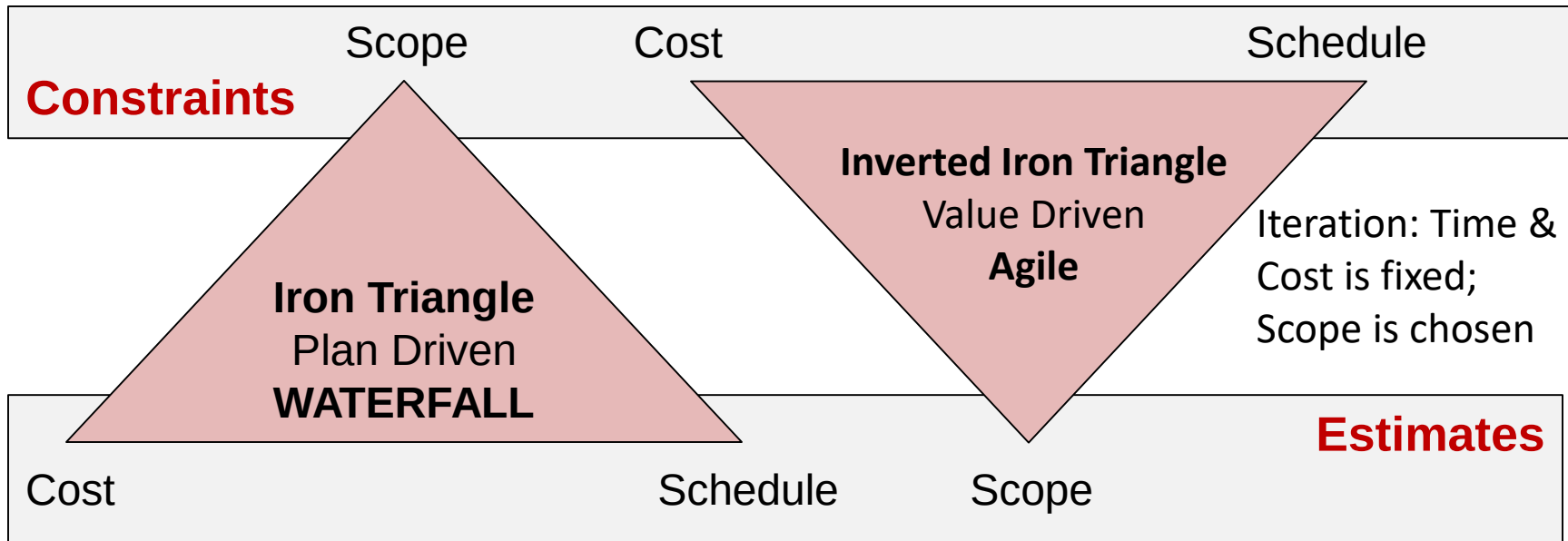


Time-boxing

- Time-boxes are short, fixed duration periods of time in which activities or work are undertaken
- Short time box alleviates the problems of
 - Parkinson's law (work expand to time available)
 - Student syndrome (11th hour seriousness)



Agile Triangle



EXERCISE – LARGE V/S SMALL BATCH

Movement of Large Batch through a system

- Divide into groups of five people, with 10 coins per group. One person is the timekeeper. The remaining four people process the coins.
- Person by person, flip all coins one at a time, recording your own results (heads or tails)
- Pass all coins at the same time to the next person
- Time keeper records time from the start of the first flip to the completion of the last flip for the group

Total 5 Iterations

Time: 20 minutes

EXERCISE – LARGE V/S SMALL BATCH

Movement of Small Batch through a system

- Similar four person process
- Each person flips each coin one at a time and records the result
- But, passes each coin as flipped
- The time keeper records the time from the start of the first flip to the completion of the last flip

Total 5 Iterations

Time: 20 minutes

3. AGILE FRAMEWORK

Birth of Agile: 2001 Feb

AGILE



AGILE ALLIANCE

Common philosophies in Light weight methodologies

Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:



While there is value in the items on the right, we value the items on the left more



Individuals and interactions OVER processes and tools



Working software OVER comprehensive documentation



Customer collaboration OVER contract negotiation



Responding to change OVER following a plan

Agile Principles

Individuals and interactions over processes and tools:

1. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done
2. The best architectures, requirements and design emerge from self organizing teams
3. At regular intervals, the team reflects on how to become more effective, then tunes & adjusts its behavior accordingly (Retrospect)
4. Agile processes promote **sustainable development**. The sponsors, developers, and users should be able to maintain a constant pace indefinitely
5. The most efficient & effective method of conveying info to and within a development team is **F2F** conversation (Co-location)

Agile Principles

Working software over comprehensive documentation:

6. Working Software is primary measure of progress
7. Our highest priority is to satisfy the customer through early & continuous delivery of valuable software
8. Deliver working Software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale
9. Continuous attention to technical excellence and good design enhances agility
10. Simplicity - the art of maximizing the amount of work not done - is essential (Design for today)

Customer collaboration over contract negotiation:

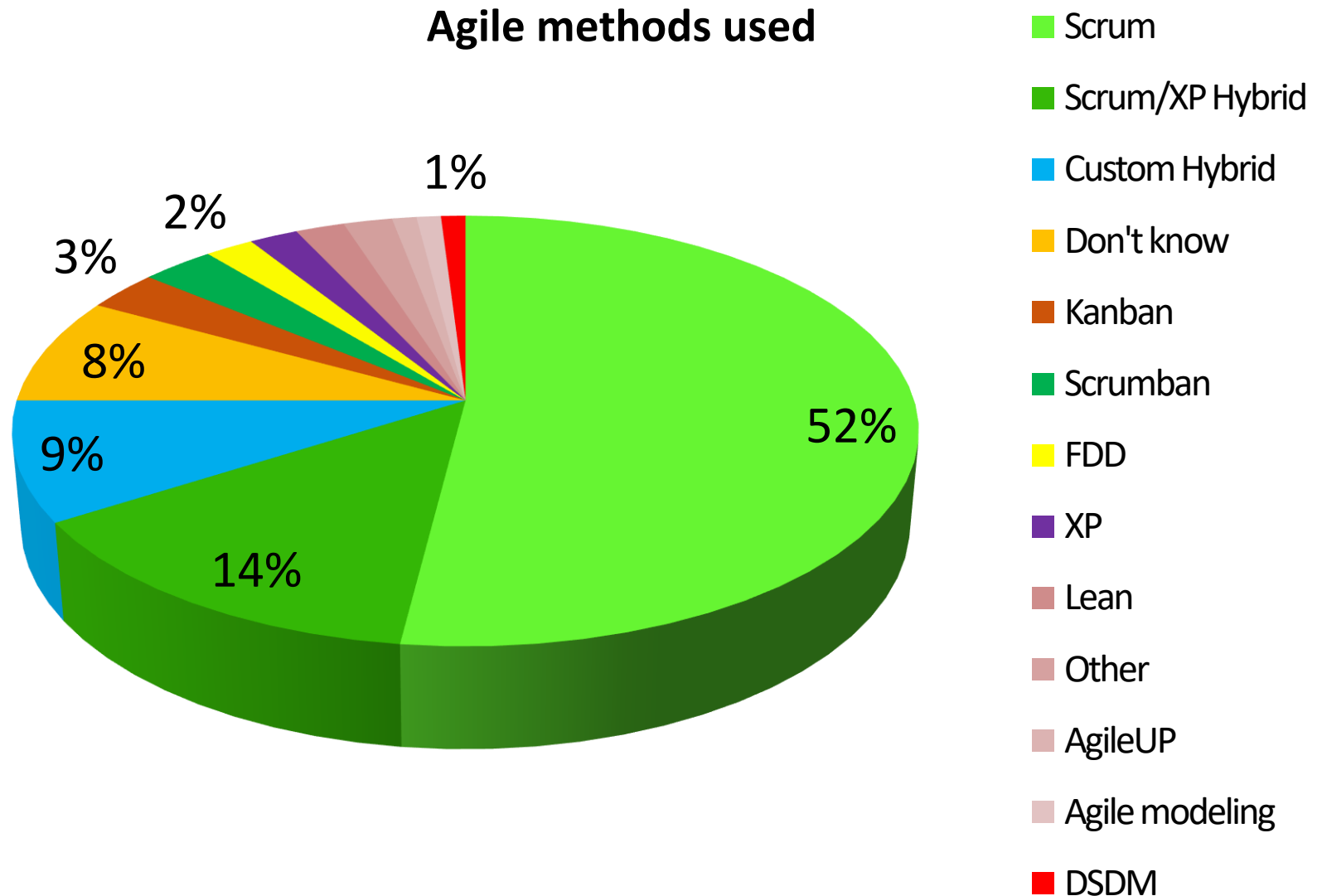
11. Business people and developers must work together daily throughout the project

Responding to change over following a plan:

12. Welcome change in requirements, even late in development. Agile processes harness change for the customer's competitive advantage

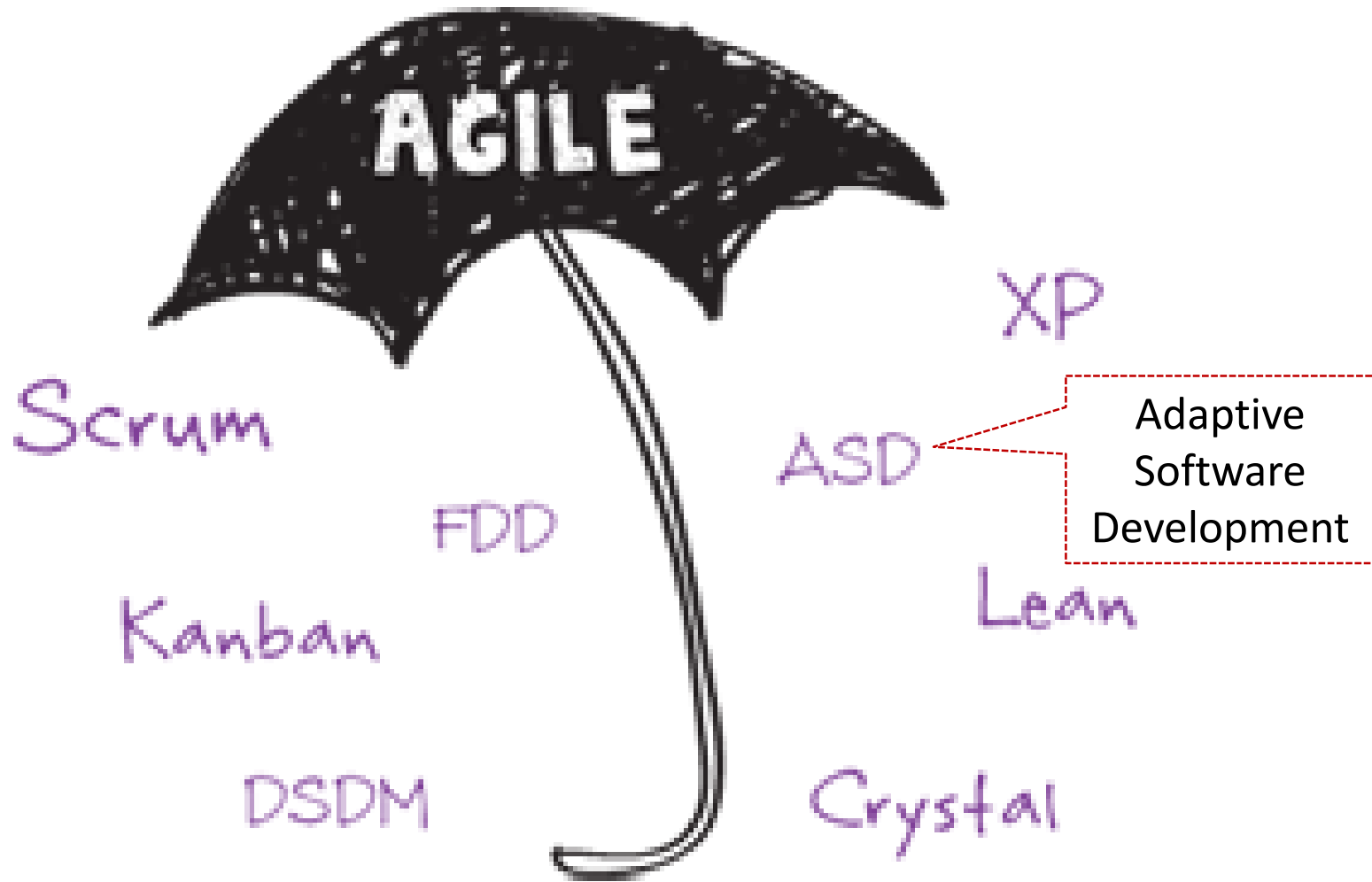
4. AGILE METHODOLOGIES

Agile Methods



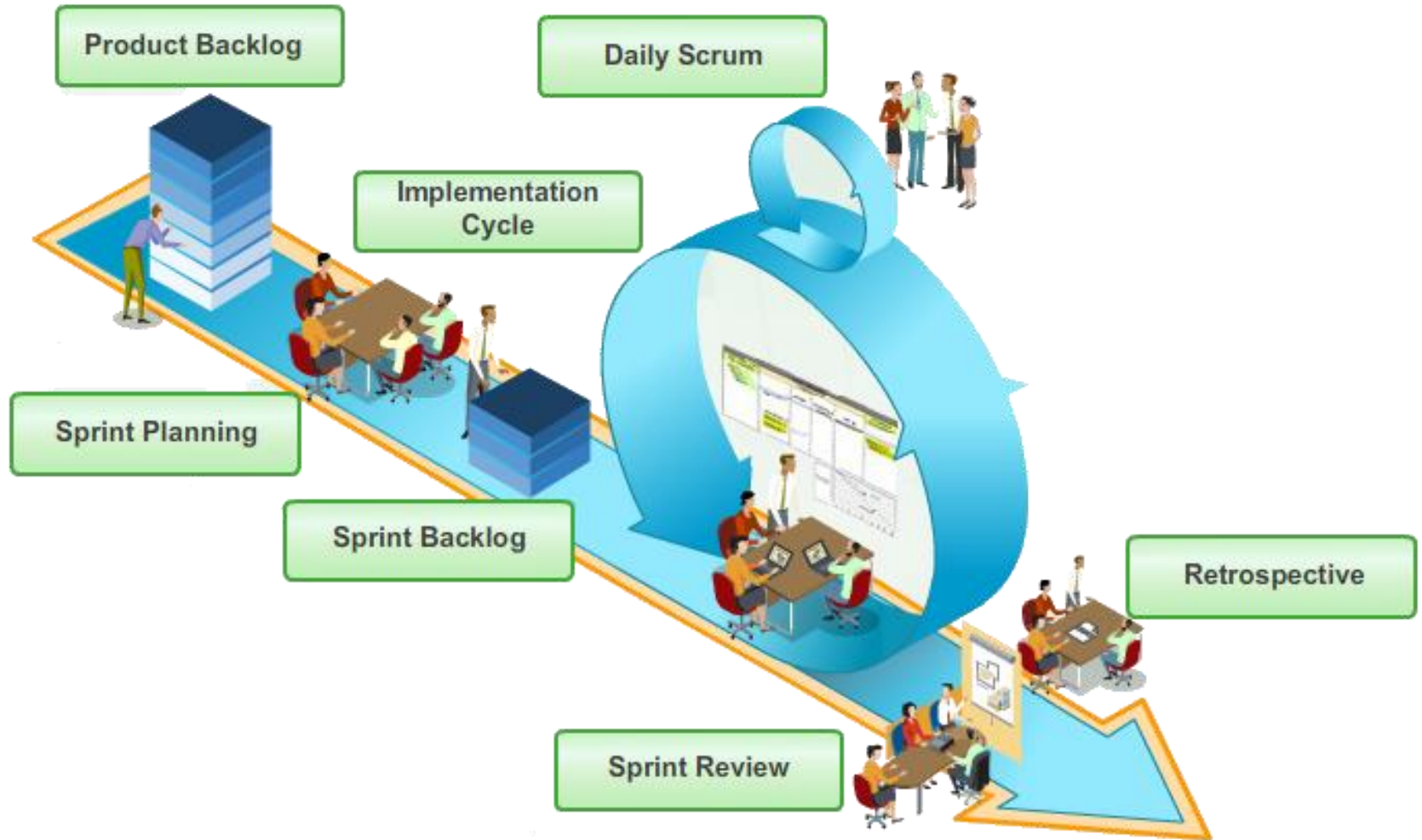
VersionOne: House.D (2012) 6th annual state of agile Survey

Agile Methods



<http://www.versionone.com/Agile101/Agile-Development-Methodologies-Scrum-Kanban-Lean-XP/>

Scrum in Nutshell



Scrum overview

- Scrum is a **structured process framework** to develop & sustain complex products
- **Scrum = $\sum$ Roles + Events + Artifacts + Rules that bind them together**
 - Using these, Scrum team iteratively builds increments of the solution, involving the customer frequently to ensure they are creating the right product
- Lightweight, Simple to understand, difficult to master
- Tobias Mayer stresses that Scrum is essentially 5 things: Self-organization, collaboration, focus, alignment, rhythm

Scrum Roles

Development Team

- Self organizing, Cross functional
- Create Increments
- Set Ground rules
- Size 3-9



Product Owner

- Voice of Customer
- Project vision
- Maximize Value
- Maintain Product backlog
- Change management
- May not be involved with funding



Scrum
Team

Scrum Master

- Scrum understood & enacted
- Working Environment
- Servant leader
 - Impediments & Interruptions
- Vision, Goals, Backlog items → DEV team

Scrum Events

#		Events	Participants	Time	Purpose
1		Sprint		1 Month*	- Immutable Sprint goal - Potentially shippable Increment
2	Inspect & Adapt	Sprint Planning	Scrum Team	8 Hours	- What (Sprint goal) & How - Sprint backlog Includes plans - Story split
3		Daily Scrum	Dev Team, SM	15 Min	- Yesterday's weather, Today's plan, Impediments - Dev. team self reporting - Sprint backlog update - Scrum of Scrums
4		Sprint Review	Scrum Team	4 Hours	Product feedback Inspect Increment & Adapt PB
5		Sprint Retrospective	Dev Team, SM	3 Hours	Process feedback Inspect & adapt processes

* Iteration length is decided by project team based on control requirements along with product owner

Scrum Artifacts

Tangible by-product produced during product development

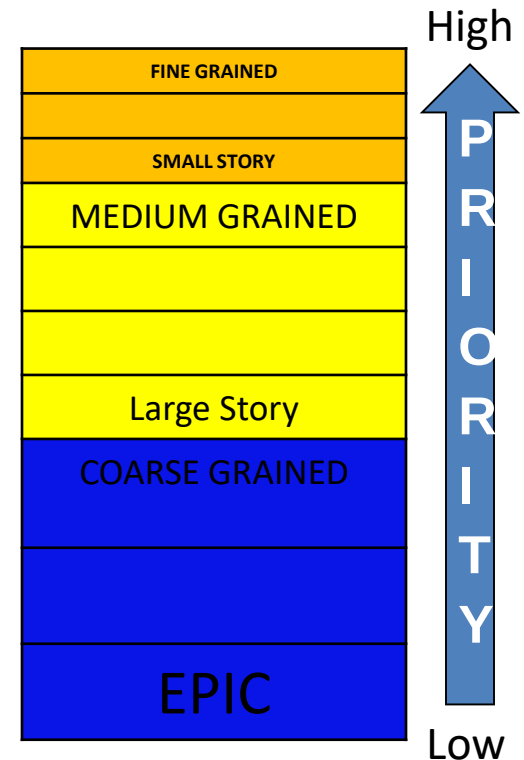
1. Product backlog

- features, functions, enhancements, fixes...
- Description, Estimate, Priority
- Priority: value, cost, risk, necessity
- **DEEP**: Detailed appropriately (rolling wave), Estimated, Emergent, and Prioritized
- Product backlog Grooming
- Only one PB even for multi teams

2. Sprint Backlog

- PB items for the sprint + Task plans
- Contains User story, Tasks, Schedule, Resources, Status, Remaining work {DAY 1, DAY 2,..., DAY N}

3. Increment = $\sum$ completed PB items



Scrum

- Iteration 0: Sprint Platform
 - Development Platform: Machine, Environment, Tools...
 - Architecture, Strategies for Test, Build, Deployment
 - Training, Project wiki
 - Procurement
 - No Code
- Sashimi
 - Agile concept of delivering value in short sprints as per 'Definition of Done' (Shippable quality)

Extreme Programming: XP

High focus on the SW Engineering Practices

- Good Practices are taken to extreme levels

Built on 5 values:

1. Communication:

- Seamless Comm., Including Customer
- F2F

2. Simplicity:

- Design for today, KISS (Keep It Simple Short)

3. Feedback:

- Product, Process, People
- Fast Failure

4. Courage:

- Break cultural & Customer barrier
- Give & Take Constructive Feedback
- Innovative

5. Respect:

- All

XP Roles

Product Manager (aka Product owner)

- Maintain & Promote Project Vision

Coach

- Realize full potential
- XP process

Domain Experts (SME)

- Set the rules

Executive Sponsor

- Fund

(Onsite) Customer

- Product: Define, Prioritize, Direction
- Release Planning

Interaction designer

- Define UI

Graphic Designer

- Define GUI

Project Manager

- How to interact with the rest of the company
- Manage external SH

Programmers

- Craft Product, Reduce cost
- Everyone Code, Test & Design
- Generalizing specialists
- Size: 4-10
- Designers, Architects, Testers, DB designer, Security expert, Network architect...

- Individual can handle multiple roles

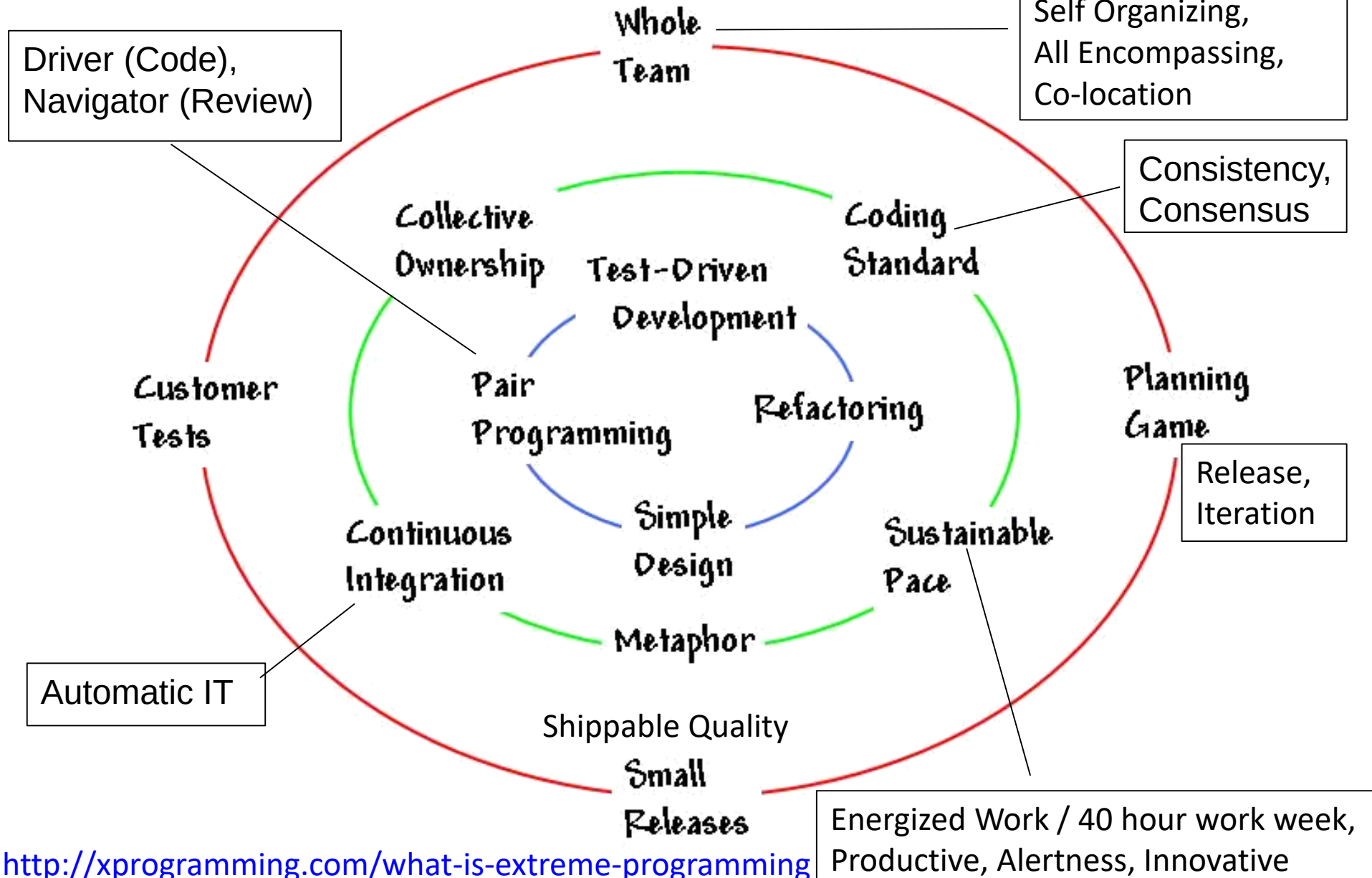
- Business Analysts: Clarify & Refine business requirements; Liaison team-customer

- XP Tracker: Measures and communicates the team's progress

- XP Tester: Helps the customer define AT, Executes AT



XP Practices



<http://xprogramming.com/what-is-extreme-programming>

Lean Software Development

Agile
“Maximize Value”

Lean
“Minimize Waste”



- Lean manufacturing
- *Optimize the whole with speed and sustainability*
- *“fast-flexible-flow” - Womack & Jones*

Tools:

- Value Stream Mapping
- 5S (Sort, Set in order, Shine, Standardize, Sustain)
- 5 Why
- Visual Control
- 7 forms of waste
- Pull systems
- WIP techniques
- Business Value Delivered Chart
 - Visual control to represent the velocity of business solutions delivered

7 Principles

1. Eliminate Waste
2. Empower the Team
3. Deliver Fast (Feedback, Maximize ROI)
4. Optimize the Whole
5. Build Quality In (QA)
 - Refactoring, automatic build & test
6. Defer Decisions: Mitigate uncertainty
7. Amplify Learning: Short Iterations

Mary Poppendieck

Tom Poppendieck

7 Forms of waste in Software

#	Waste	Example
1	Partially done work	Code waiting for QA, Specs waiting for development
2	Extra processes	Unused / over documentation, unnecessary approvals, bureaucracy
3	Extra features	Less / no value adding for end user, gold plating, Technology showcase
4	Task switching	People assigned multiple projects
5	Waiting / Delay	for reviews, approvals, clarification, guidance, dependency (internal / external), feedback, slow communication...
6	Motion	Info / deliverable movement from one to another Distributed teams, hand-off
7	Defects	Defective documents, requirements or software

Mary Poppendieck

Kanban Development

- Japanese word: Billboard, Signboard
- Continuous delivery without team burnout
- Focus on cycle time optimization for best throughput
- 5 core Principles
 1. Visualize the workflow
 - Organize, Optimize, Track
 2. Limit WIP
 - to avoid over commit, inefficiency, bottleneck masks
 - Minimize waste (rework), investment capital, cost
 3. Manage Flow
 4. Make Process Policies explicit
 5. Improve Collaboratively
 - Models & Scientific
 - Improve in Small steps

Kanban Development

Backlog	Selected (4)	Develop (3)		Acceptance (2)	Deploy
		Ongoing	Done		
P	Q	A	C	T	V
L	R	B		U	
M	S				
N					
Z					

KANBAN BOARD with WIP limit (Pull System)

- Maintenance Projects
- Speed (Backlog-Complete) SLA
- Work-in-progress limit -> Predictable throughput
 - Smoothens the average lead time (Less variance)
 - Agreements such as "we agree to deliver new features within 21 days with a 95% confidence limit" are possible

DOING AGILE



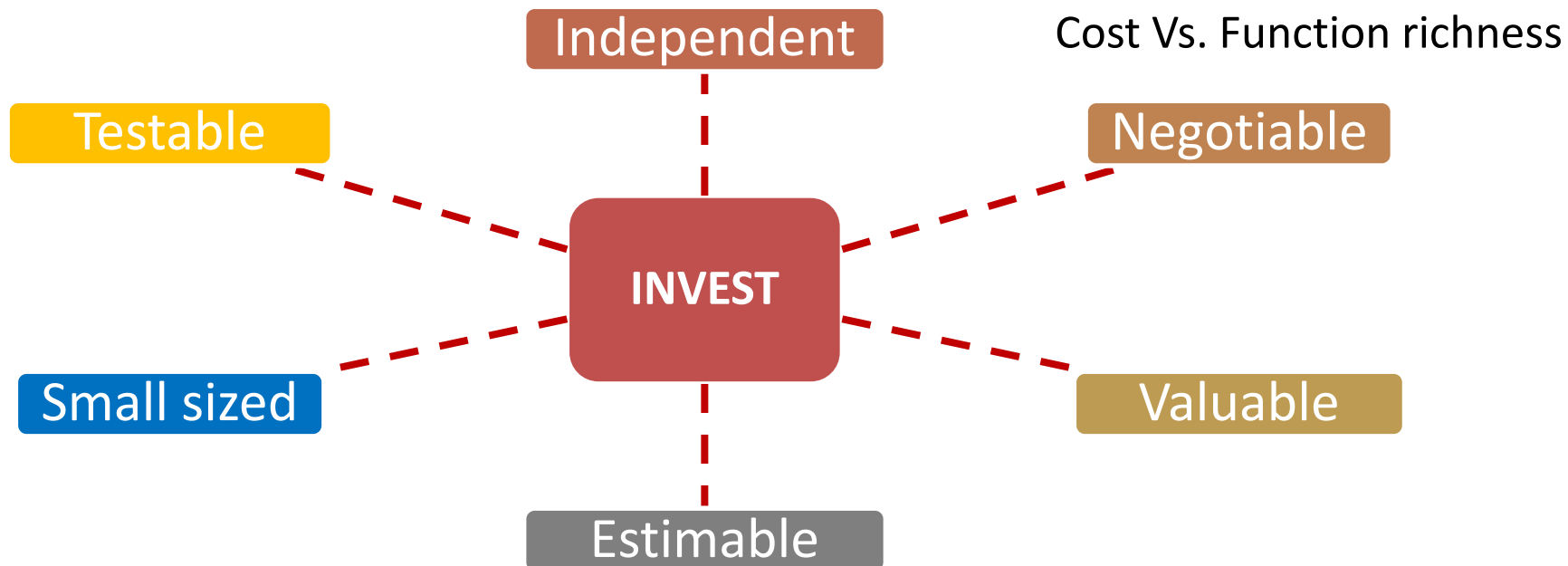
5. CEREMONIES

User Stories

- Short, simple description of user wants in user/customer language
 - Typically written by Customer
 - Acceptance criteria is also captured
- Small enough to fit in an iteration; else split
- Format: *As a <ROLE> I want to do <FEATURE> so that I can do <REASON/BENEFIT>*
 - i.e., WHO wants WHAT & WHY. WHY is optional, but recommended
 - “As a frequent traveler, I want to reserve a hotel room”
- Another format, often used for nonfunctional or system based requirements is ‘Given .., When ..., Then ...’
 - “Given user account is valid, When user password age > 3 months, Then force password change”

User Stories

- Types
 - Business/Functional, Non-Functional, Bug, Spike, documentation, technical...
- INVEST
 - Small, independent, valuable chunks of work-functions of which can be negotiated with the business to find the right level of cost Vs. performance; and we must be able to readily estimate and test these pieces of work.



User Stories: 3 C's

User stories follow the XP alliteration of the '3C'

1. Card: Short statement of intent on a card

As a user, I can login & gain access to the intranet, so that I can collaborate with all the organization

2. Conversation: Discussions

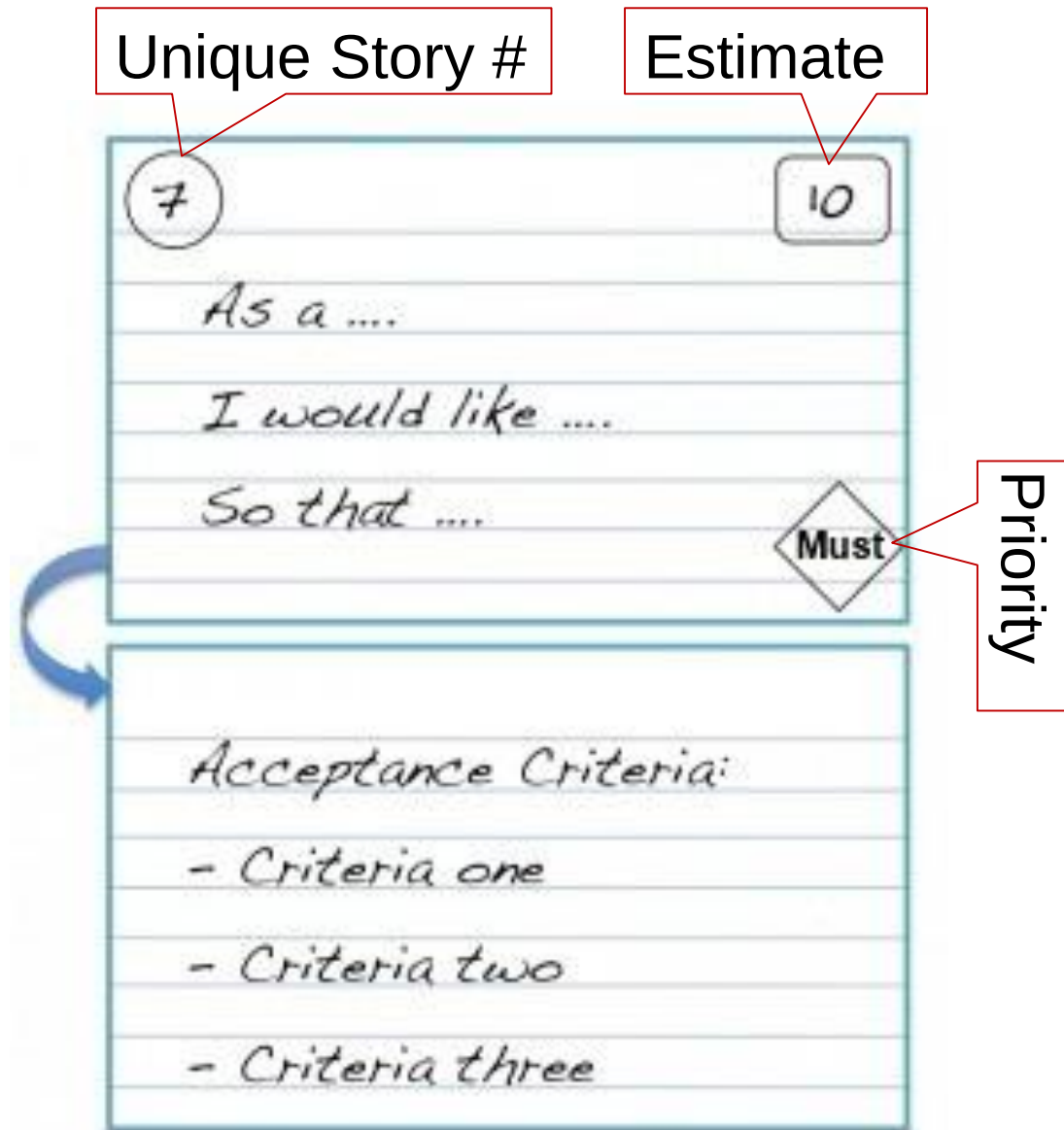
What about expired accounts?

Can it remember my login?

3. Confirmation: Acceptance criteria

- Expired accounts fail
- Remember the login, not the password
- Account lockout in 3 consecutive wrong attempts

User Stories in Index Cards

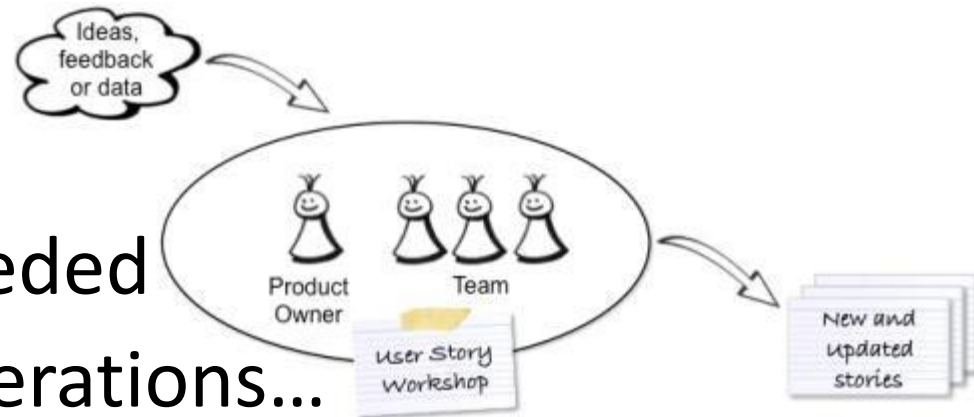


Agile Requirements

- Epic: Big User Story
 - Complex: Difficult to disaggregate to constituent stories
 - Compound: Comprise of multiple short stories
- Theme: Collection of related user stories
- Task: Decomposition of user stories for estimation, planning, execution & tracking
- Trawling techniques:
 - Interviews, Focus groups, Facilitated workshops, nominal group technique, prototyping, wireframes, personas...
- Disaggregation:
 - Splitting a story or feature into smaller, easier-to-estimate pieces

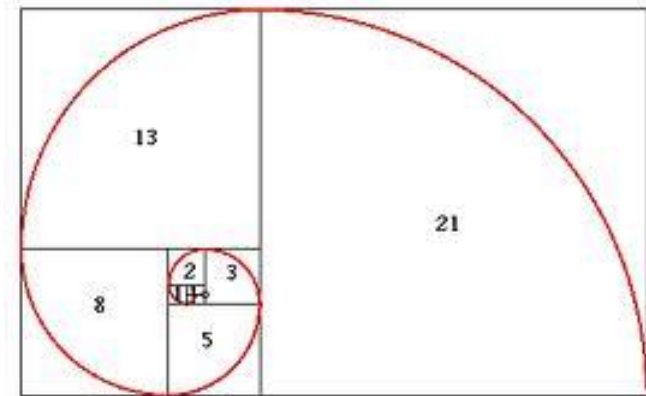
Story writing workshops

- Facilitator + Scrum team + any SH who can contribute
- 2-4 Hours
- Focus on WHAT is needed
- Brain storming, Deliberations...
- Work around Themes, Features, Epics
- Begin with the Vision / end in mind
- Write as many stories as possible
 - Include acceptance criteria
- Conducted as required



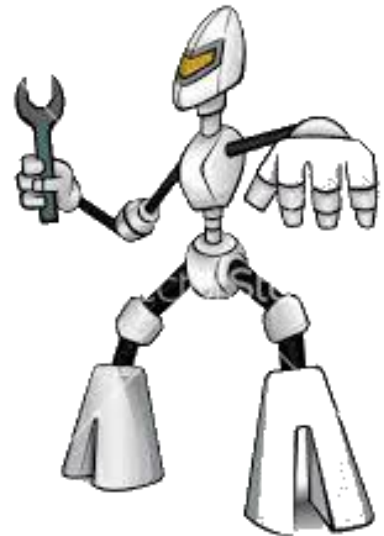
Relative sizing / Story points

- Relative estimation is easier, quicker, accurate
- No Units
- Does not directly co-relate to actual effort
 - Removes 'Time' weird connotation
- Baseline story & Triangulation
- Estimate is all inclusive
 - Testing, Complexity, Effort, Risk
- Scale: Fibonacci, T-Shirt size...
- Not comparable across projects



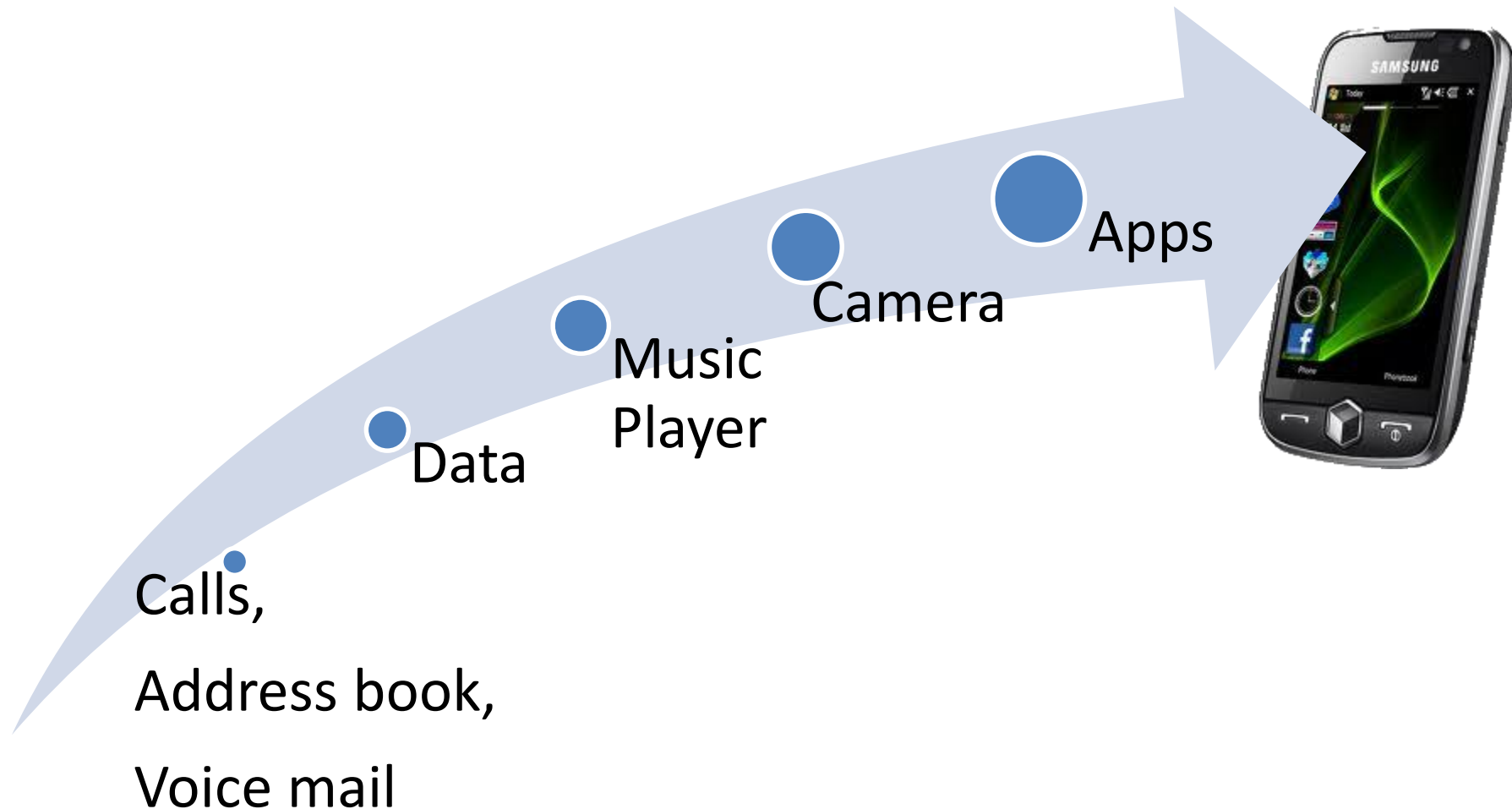
Ideal Time

- Time take to complete a story uninterrupted & without waiting time !!
- **Elapsed Time, Real time, Actual time, Calendar time:** The actual amount of time it will take to complete the story
- Convert to 'elapsed time' estimate based on velocity / historical data



Minimum Marketable Feature: MMF

MMF represents package of functionality that is complete enough to be useful to users or market, yet small enough that it does not represent the entire project

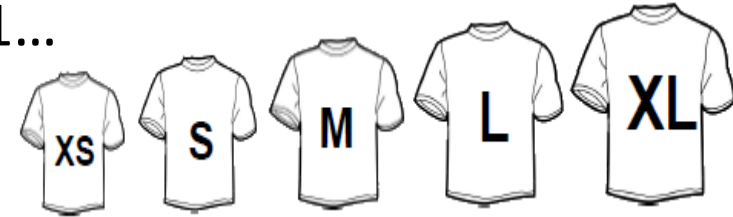


Wide Band Delphi

- User story estimation
 - Iterative – Adaptive - Collaborative
 - Mitigates Bandwagon & Halo effects
1. Define the problem
 2. Agree on (revised) assumptions
 3. Independently estimate (bottom-up)
 4. Facilitator collates & shares the summary with new assumptions
 5. Iterate 2-4 till exit criteria

Planning Poker

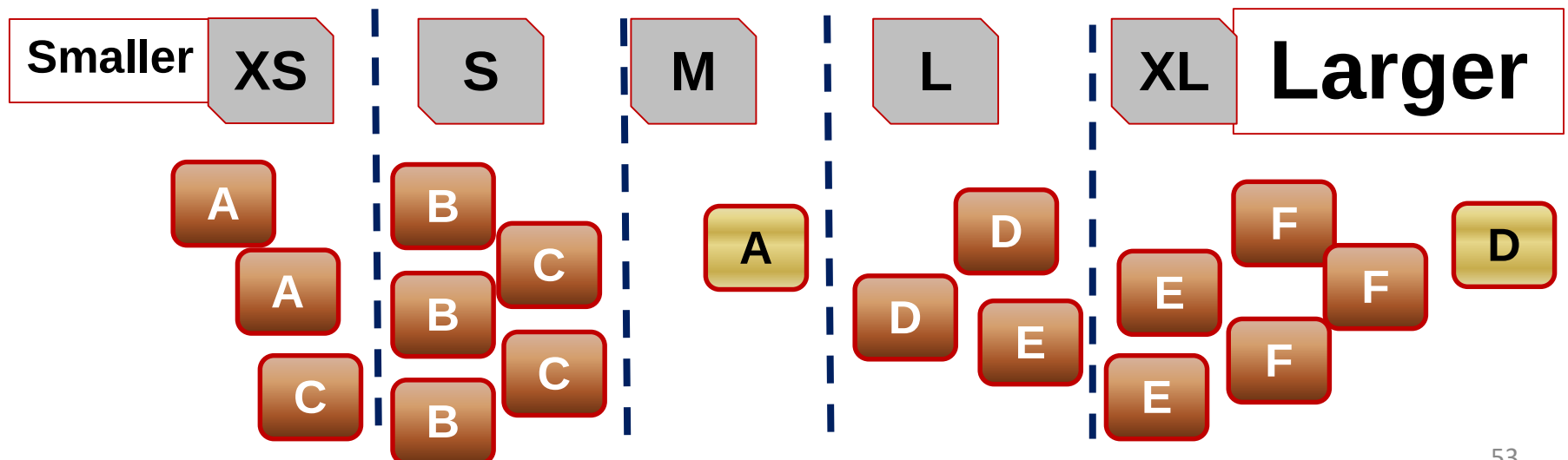
- Wide Band Delphi in Practice
- Deck of cards with distinct estimation values
 - Fibonacci series: 1, 2, 3, 5, 8, 13, 21...
 - T-Shirt size: XS, S, M, L, XL



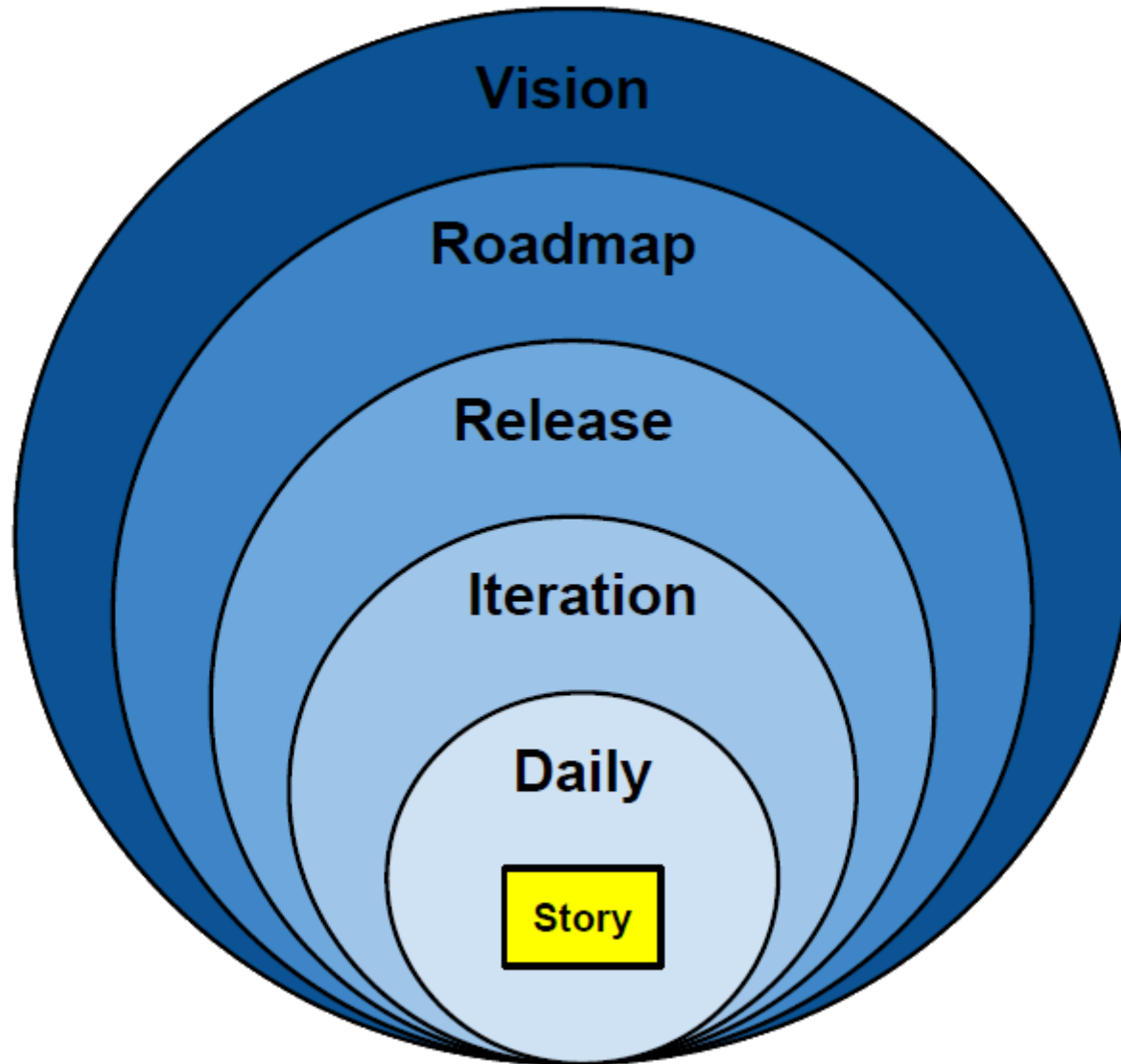
1. PO reads & explains the story
2. Independently triangulate estimate & disclose together
3. Discuss differences, esp. outliers, and re-estimate until estimates converge
4. If no convergence after few iterations,
 - PO intervene and clarify doubts to estimator(s)
 - Tie-breakers: Majority, High, Average, 3 point (PERT) estimate

Affinity estimating

- Product backlog (Min 20, Best 40) estimation
- A form of triangulation
 1. Silent Independent relative sizing between Smaller-Larger
 2. Editing of Wall – Consider others' estimate
 3. Place items into relative sizing buckets like T-Shirt size
 4. Product Owner can 'Challenge'
 5. Record the estimation



Agile Plans Evolution

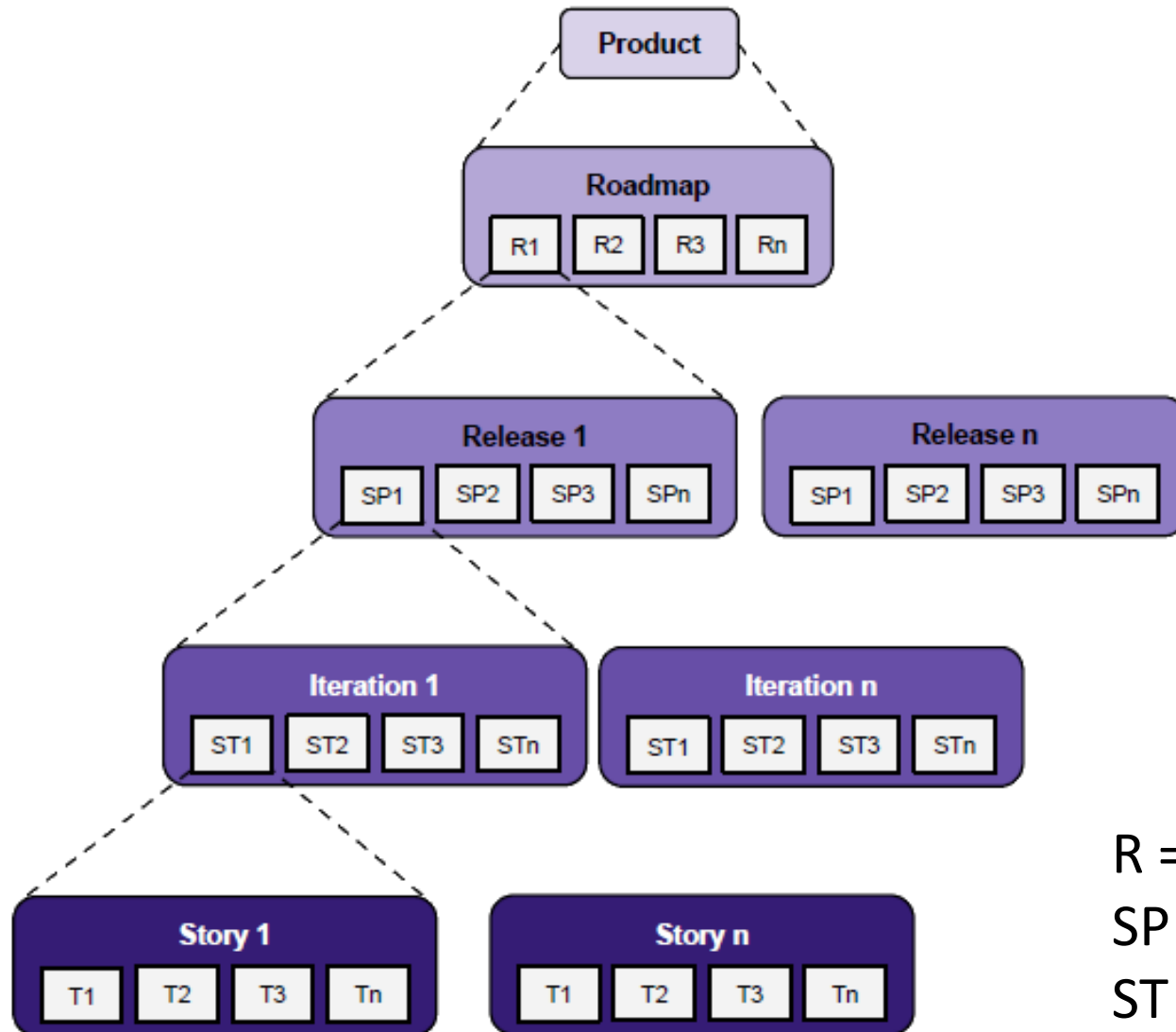


Project Vision

- Vision of the product, High level efforts, Target customers, USP
- Project Vision Meeting
 - Level set on Project understanding
 - Goals, Assumptions, Risks
- Elevator statement
 - Short (~ 2 min) Statement on Justification
 - Not discuss who, what, when, or where
- Project tweet:
 - Project Goal in 140 characters or less



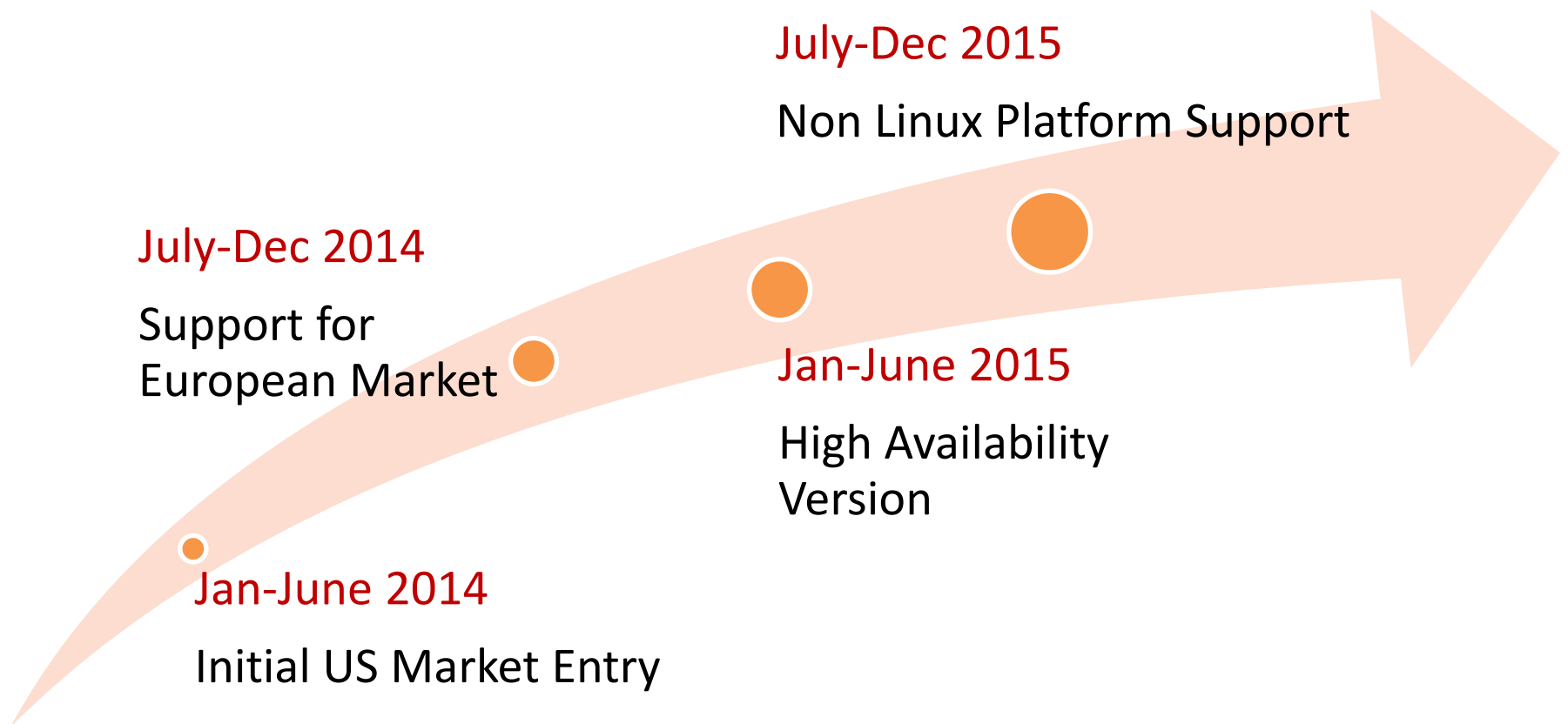
Agile Plans



R = Release
SP = Sprint
ST = Story
T = Task

Product Roadmap

- Visual overview of Product releases and its major features
- Consider MMF
- Story-map tools can help



Release Planning

- Release
 - Delivery of new functionality to end user
 - Scope or Time driven
 - Set of iterations, spanning 3-12 months
 - Constrained by competitor deadlines, contract, customer deadlines, MMF, team velocity...
- Release plan
 - Owned by PO; Planned with team;
 - Story maps can be used
 - Usually updated at the end of iteration
 - Release goal lead to Release backlog

Iteration Plan

- Time boxed to 8 hours for 1 month sprint
 - Developed by team; Output is Sprint backlog
1. Set Sprint goal with PO
 2. Selects the high priority user stories for the sprint based on velocity & availability; Understands; Splits as necessary
 3. Decompose story to tasks as per the 'definition of done' (WHAT)
 4. Estimate the tasks, typically in time-units
 5. Derives schedule based on Project / Resource calendars (WHEN, Optionally WHO)

The PO has the final say on the priorities for the sprint, and the development team has the final say on the amount of work that can be accomplished in the sprint

Sprint backlog

		Remaining Work (hours)						
User Story	Task	Day 1	Day 2	Day 3		Day N	Who	Status
A	Code the user interface	8	4	0	.	0		
	Code the middle tier	16	12	3	.	0		
	Write the online help	0	4	8	.	0		
	Add error logging	0	8	8	.	0		
	Test	8	16	8	.	0		
B	Code class Boo	0	0	10	.	4		

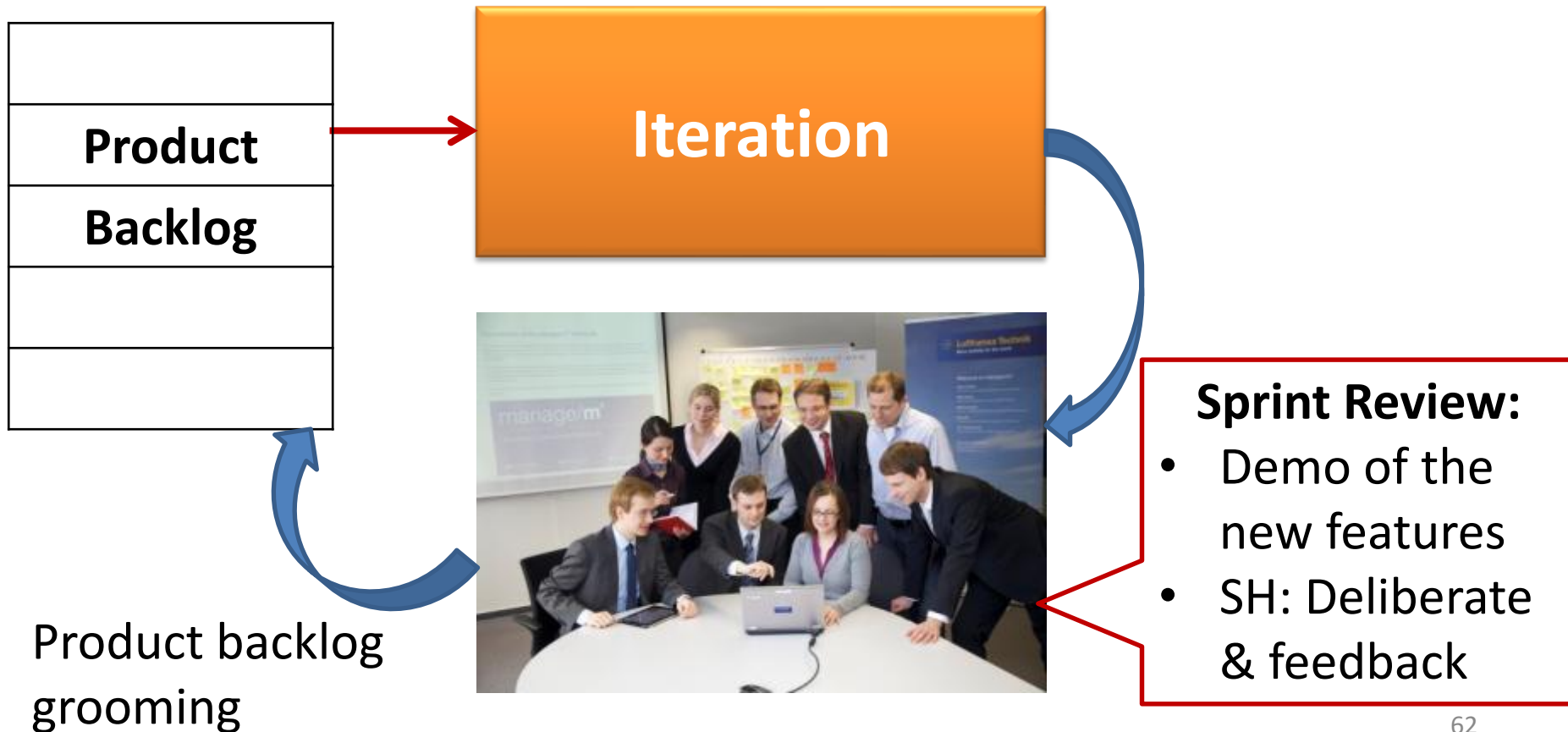
Daily Plan

- Happens as part of Daily stand up meeting
- Team share their daily plans
- Occasionally, Team can drop any user story, if it is not feasible in current sprint



Sprint review / demo

- Time boxed to 4 hours for 1 month sprint
- Inspect & adapt:
 - Aid right / innovative shaping of the product

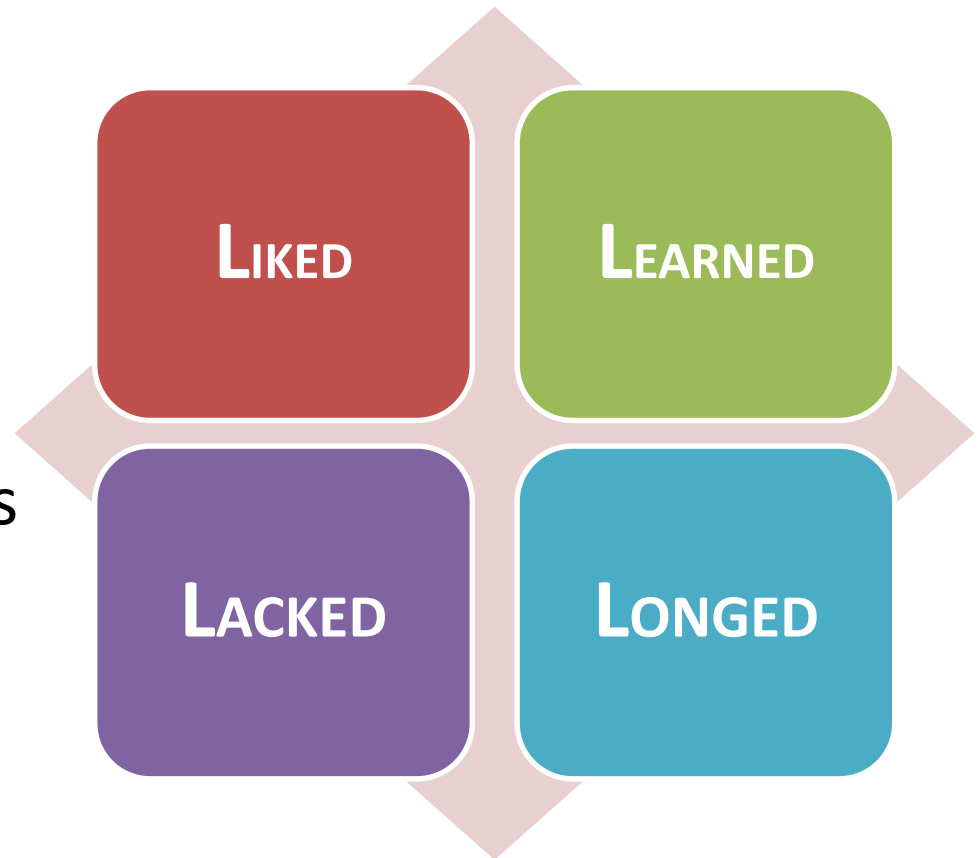


Retrospectives

- Time boxed to 3 hours for 1 month sprint
- Process Feedback for improvement of
 - Productivity (Knowledge sharing, Process efficiency, Best practice)
 - Quality
- Types:
 - Project Retrospectives
 - Release Retrospectives
 - **Iteration Retrospectives**
 - Surprise Retrospectives

Retrospectives: Agile games

- To break monotonous, 'no action taken', 'non participating lot' retrospectives
- To encourage full team participation, improve quality...
- 4L:
 1. Individual add
 2. Group discuss
 3. Decide action Items



Retrospectives: Agile games

- Sail boat / Boat / Speed boat
- Identify Sails & Anchors – Cluster – Prioritize – Summarize - Track
- Can be used in Risk identification also

SAIL

Aid move faster

START / CONTINUE DOING

ANCHORS

Inefficiencies, non/less value adding...

STOP DOING





EXERCISE – USER STORIES & ESTIMATION

USER STORIES

Each group takes up a real work related problem (please keep it simple)

Now work as a team to break down the problem into user stories and give each Story an ID Number.

Also create an Acceptance Criteria for each story

Use one of the format below for User Story including Title & Narrative.

Narrative should include one of these standard formats:

As a <Role>, I want to <Action> so that <Benefit>

A <Role> can <Action> so that <Benefit>

<Role><Action> so that <Benefit>

Swap with neighbor, review for clarity, and revise

Time: 20 minutes

PLANNING POKER

1. Nominate a Product Owner
2. Pick Scrum Master to facilitate the estimation session
3. Scrum Master will call out a story for estimation
4. Each team member, based on **Effort** needed to implement the story will provide it with a T shirt size XS, S, M, L or XL
5. The lowest & highest value estimators are given a chance to share their views
6. Discuss differences and re-estimate until estimates converge
7. In case of no convergence after few iterations, PO intervene and clarify doubts to estimator(s)

Tie-breakers: Majority, High, Average

Remember: Effort and Duration are not the same

Estimation is based on complexity of Effort related to volume of work, known and the unknowns

Time: 20 minutes

6. ARTIFACTS

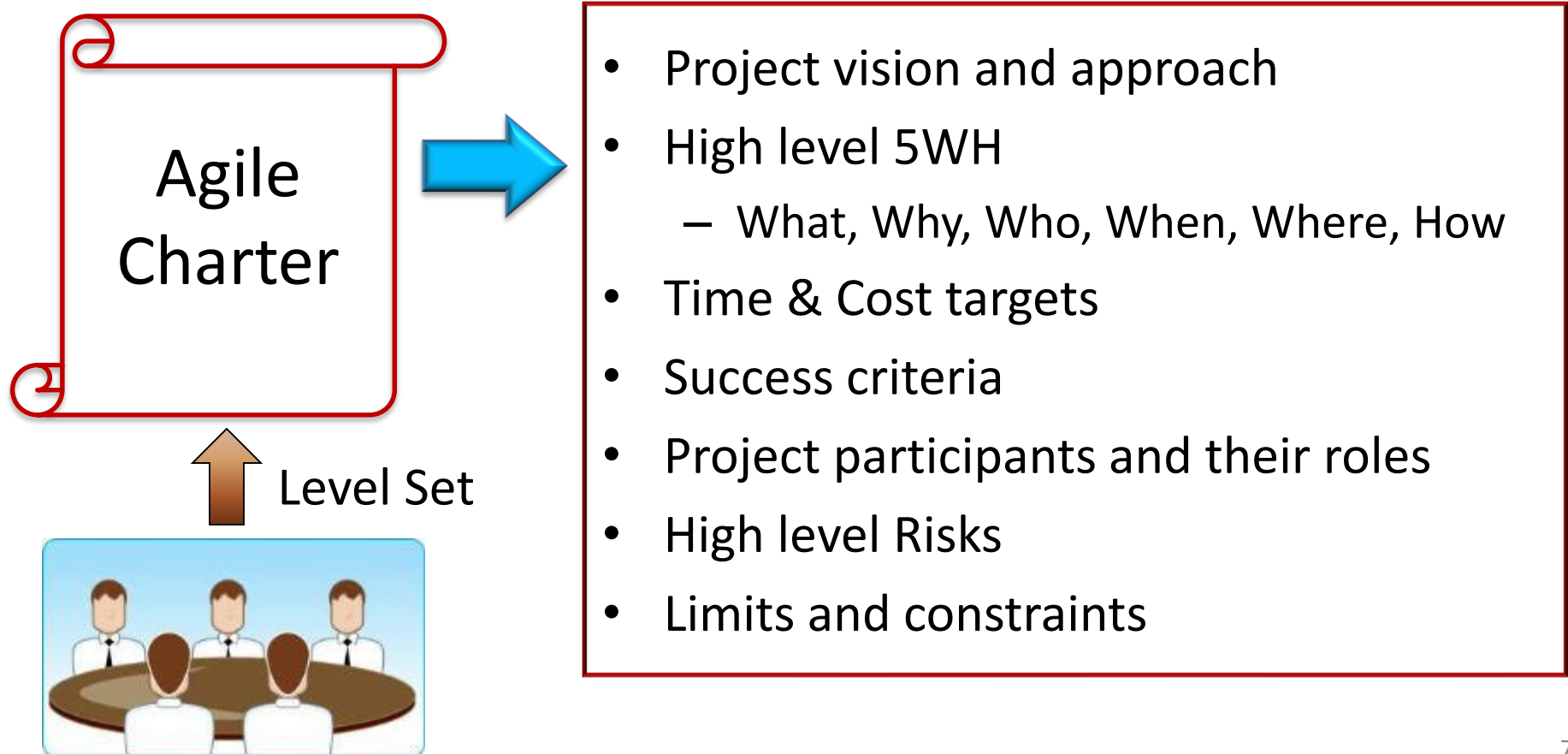
Business Case



- Project Vision
- Anticipated costs
- Anticipated benefits: Revenue, cost savings, better service, improved quality, compliance, improved working conditions, increased employee satisfaction...
- Financial analysis: ROI, IRR, NPV...
- Risk of NOT undertaking the project
- Project Risks
- SWOT & PESTLE (Political, Economic, Social, Technical, Legal, Environment) analysis
- Recommendations: **Mandatory, Critical, High, Medium, Nice** to have project

Project Charter

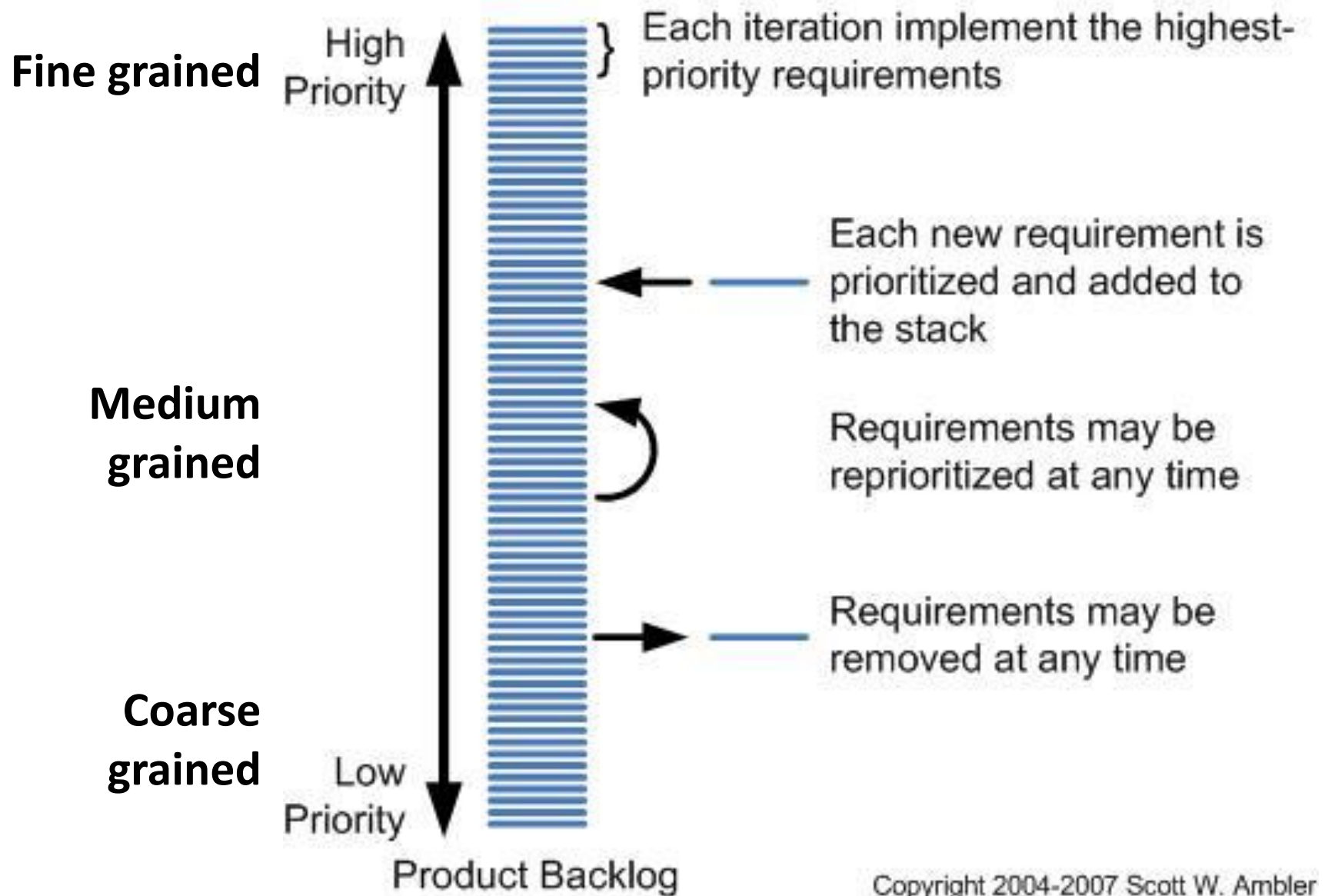
- Formal authorization to start the project
- Less Concrete Requirement
- Effective Change Management
- Early Value delivery



Product backlog

- Owned by Project backlog
- Business/Functional, Non-Functional, Bug, Spike, documentation, technical...
- Ordering: value, cost, risk, priority, necessity
- **DEEP**: Detailed appropriately (rolling wave), Estimated, Emergent, and Prioritized
- Only one PB even for multi teams
- Product backlog Grooming

Product backlog



Copyright 2004-2007 Scott W. Ambler

Product backlog

ToDo List

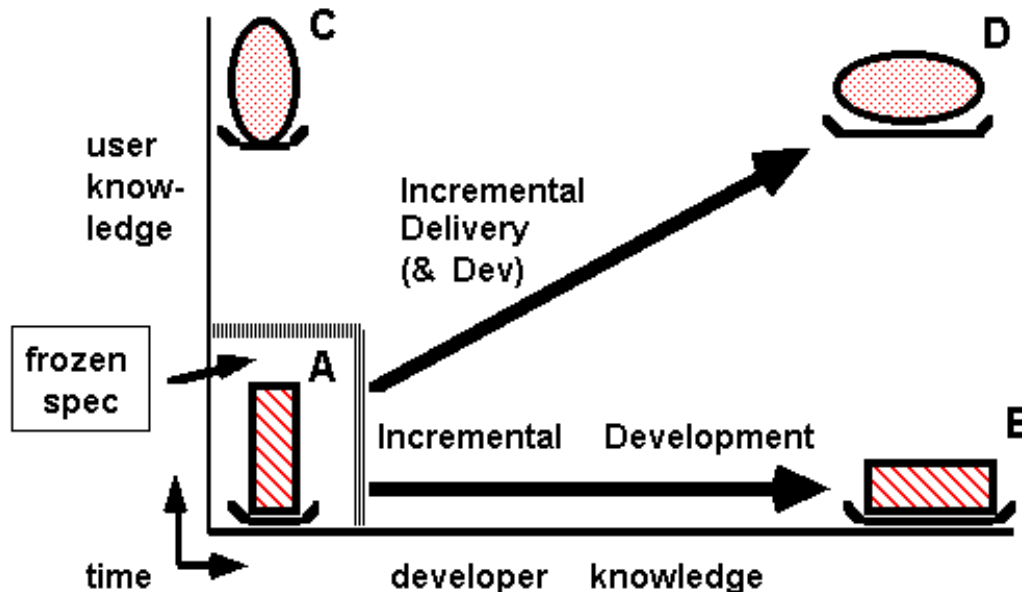
ID	Story	Estimation	Priority
7	As an unauthorized User I want to create a new account	3	1
1	As an unauthorized User I want to login	1	2
10	As an authorized User I want to logout	1	3
9	Create script to purge database	1	4
2	As an authorized User I want to see the list of items so that I can select one	2	5
4	As an authorized User I want to add a new item so that it appears in the list	5	6
3	As an authorized User I want to delete the selected item	2	7
5	As an authorized User I want to edit the selected item	5	8
6	As an authorized User I want to set a reminder for a selected item so that I am reminded when item is due	8	9
8	As an administrator I want to see the list of accounts on login	2	10
Total		30	

You can add columns for Type (Story, Risk action, Spike), Theme, Feature, Short name, Business value, Cost of Implementation, Risk, Status

Incremental delivery

Requirements are gradually established through interaction between users and developers

- Early & Frequent Release
- Potentially shippable software
- Team gets Feedback for right shaping
- Customer gets better ROI, lower TTM



EXERCISE - ARTIFACTS

BACKLOG, GROOMING & DEPENDENCY MAPPING

1. Create a backlog of the Estimated User Stories
2. Prioritize the backlog based on
 - a) Necessary sequence for Implementation
 - b) Fastest Value to Customer
 - c) Highest Risk factor for Implementation
3. Map External and Internal Dependencies for each Story

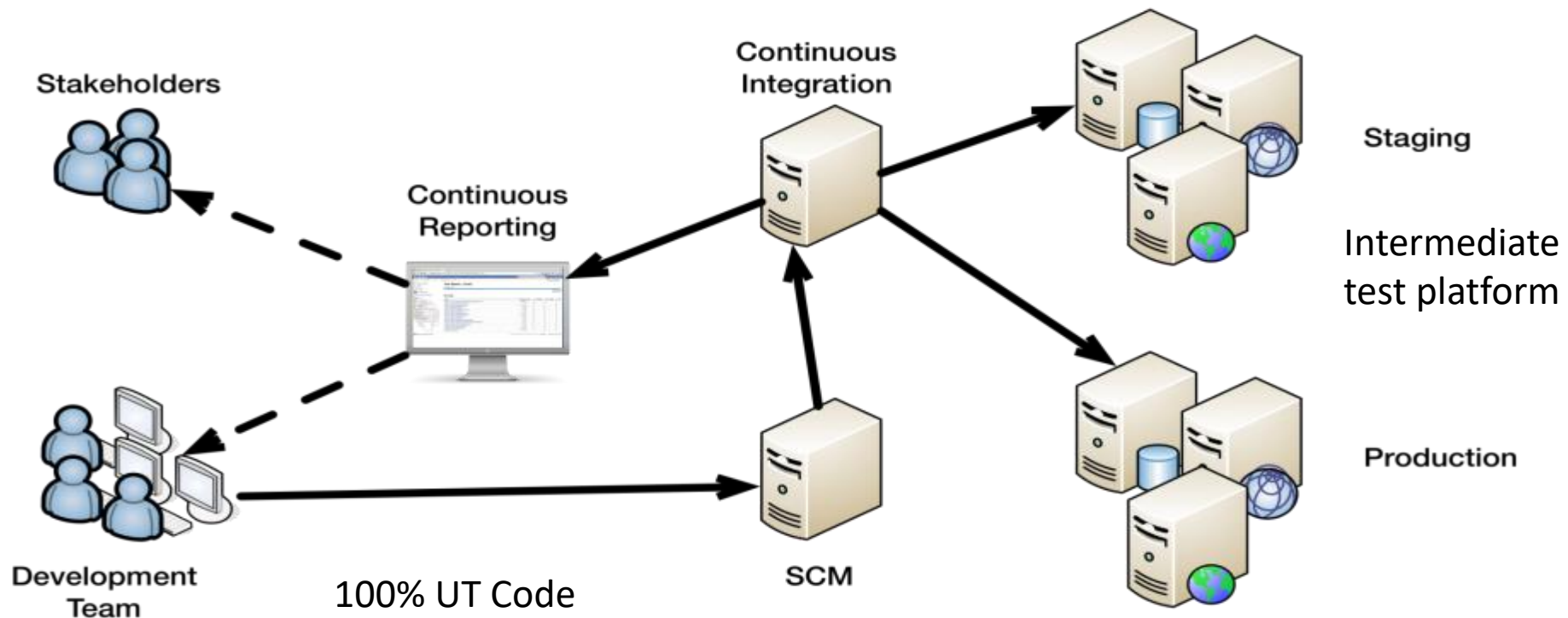
Time: 20 minutes

7. ENGINEERING PRACTICES

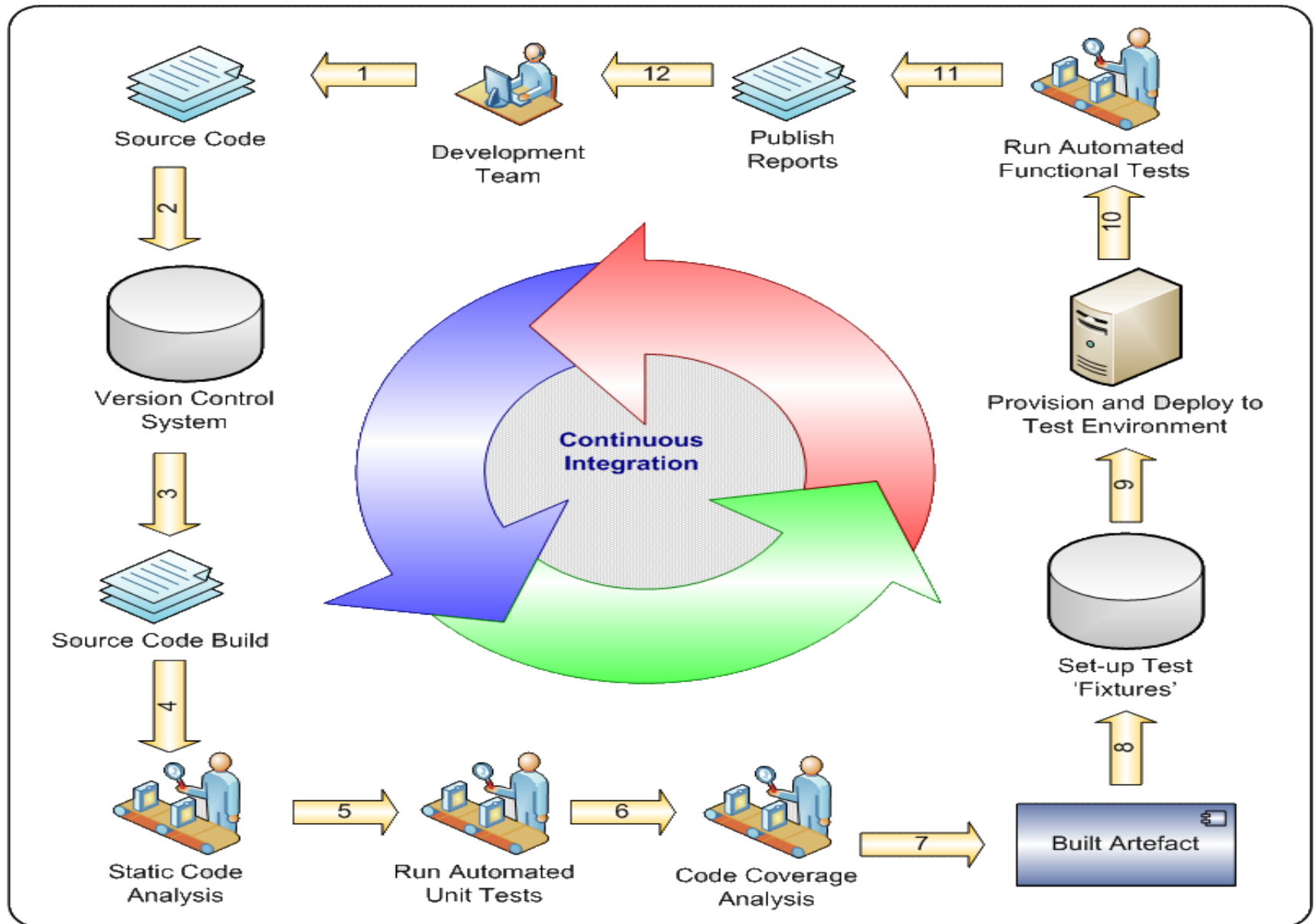
Continuous Integration (CI)

- Is a process that automatically integrates software changes, verifies the quality of these changes by running tests and reports errors immediately
- Reduces Defect cycle time

Continuous Integration Workflow



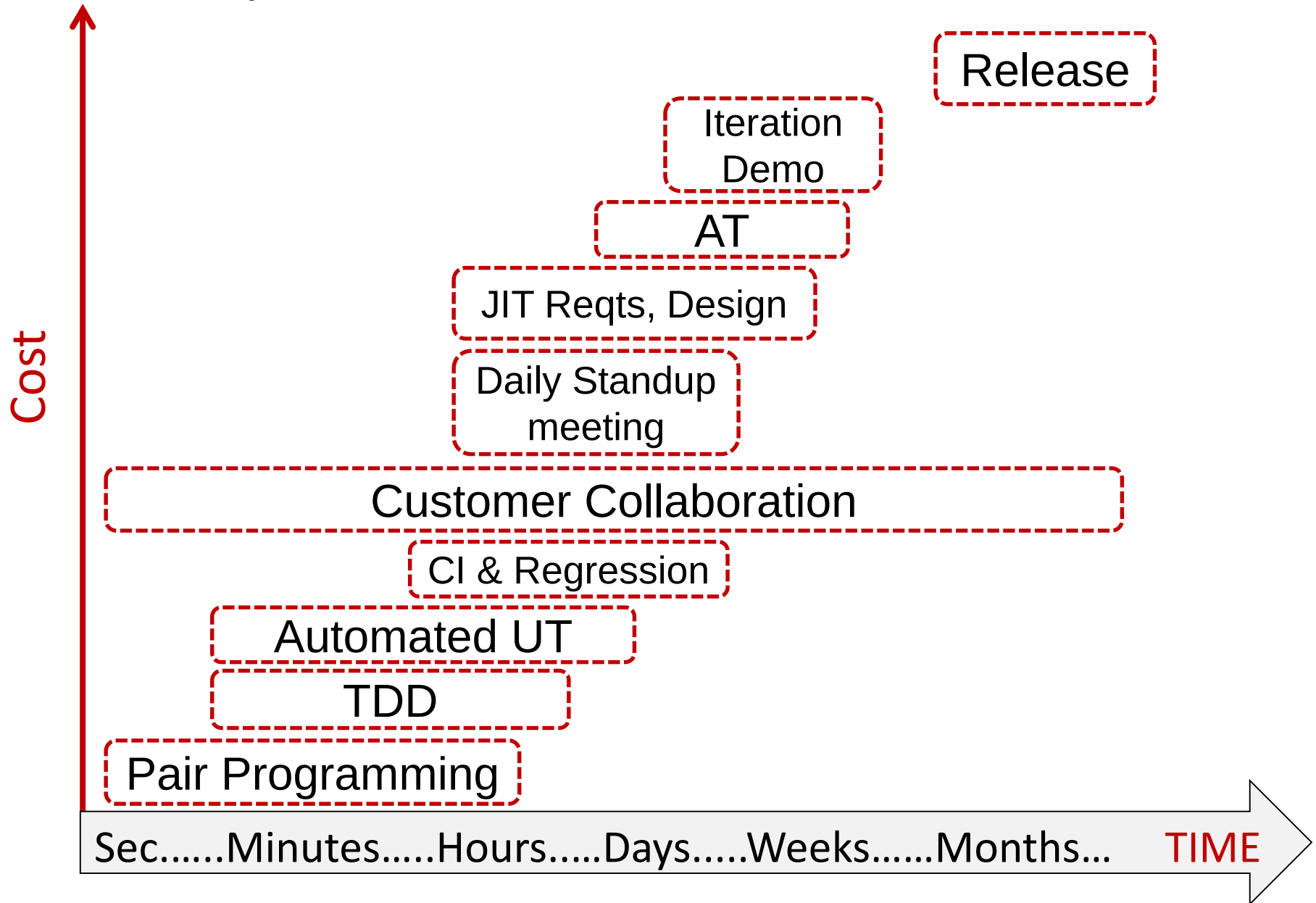
Continuous Integration (CI)



Frequent Verification & Validation

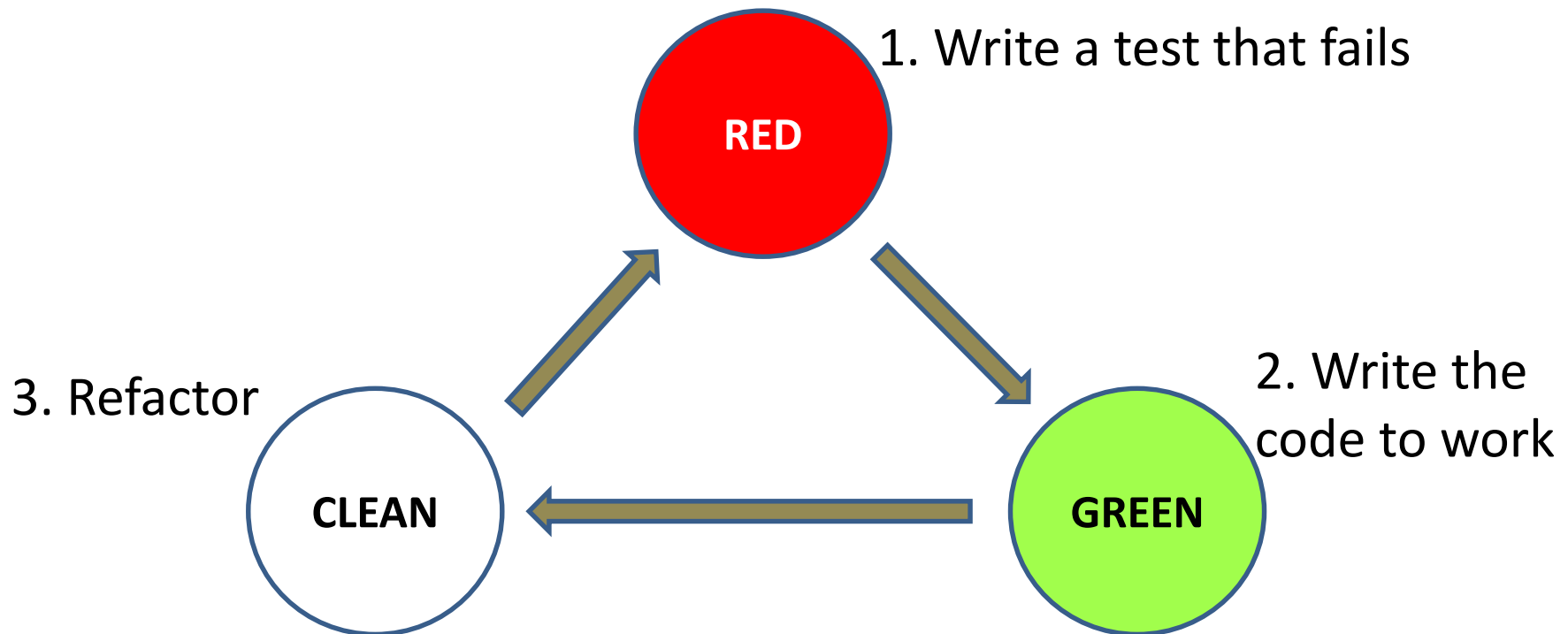
- To address
 - Human mistakes & lack of skills
 - Severe communication challenges; esp. in, Unique IT intangible projects
 - Lesser defect cycle time & Cost of change

Frequent Verification & Validation



Test Driven Development (TDD)

- 'Red, Green, Clean' or 'Red, Green, Refactor'
- Typically Test cases are automated
- Unbiased test cases
- TFD = TDD - Refactoring



Acceptance TDD (ATDD)

- Focus on business requirement
 1. **Discuss** the story and acceptance criteria together
 2. **Distill**: Automate the acceptance test cases
 - Few tools are FIT, Fitnesse (GNU), Concordian (Open source, Java), Robot framework
 3. **Develop**: Red, Green, Refactor
 4. **Demo** to business Stakeholders

8. AGILE PRACTICES

Value

- Business Value = Perceived benefit – Perceived Cost
 - The ultimate purpose of business
- Value is what customer is ready to pay for
- Maximize Value; Minimize Risk & Wastages
 - Lean, Most Valued features first, Early MMF to market
 - Fast Failure
 - Risk is anti-Value
- Frequent customer product feedback to ensure right / innovative shaping
- Agile is Value driven
 - Value decides priority which decides order of delivery, but ensure technical dependencies & risks are given due consideration



Value based analysis

Value based analysis is the process of considering the business value of work items and then acting accordingly at every stage of the project

- For Product backlog priority, consider
 - Business value
 - Frequency of use: One time → Always used
 - Cost of development
 - Risk
 - Dependency
 - Complexity, Time, Conformance, Re-usability...

Customer Valued Prioritization

- Business **Value, Risk, Cost**, Dependency, Time, Technical, MMF, Frequency of usage, Conformance, Difficulty, Re-usability...
- Prioritization Schemes to establish priority rank
 1. Qualitative Ranking: High, Medium, Low
 2. Forced Ranking Technique: Unique numbers to each requirement
 3. Monopoly Money: Budget to distribute across features
 4. 100 Point Method: 100 points to distribute across features
 5. MoSCoW Prioritization:

Must have	Should have
Could have	Would like to have, but not this time

Relative Prioritization

- Clear prioritization & ranking among all types of Items in the Product backlog
 - Features (Epic, Theme, Story), Defect, Spike, Documentation, Risk action Item...
 - Single Product backlog, even for virtual teams
- Uses
 - Deliver MOST values first
 - Best trade off during changes

Agile games

- Also known as collaborative / innovation games
- Facilitated workshop techniques
 1. Remember the feature: For Vision setting and Requirements elicitation exercise
 2. Prune the product tree: To gather and shape requirements
 3. Speed boat: Retrospective, Risk identification
 4. 4L: Retrospective

Agile games- Remember the future

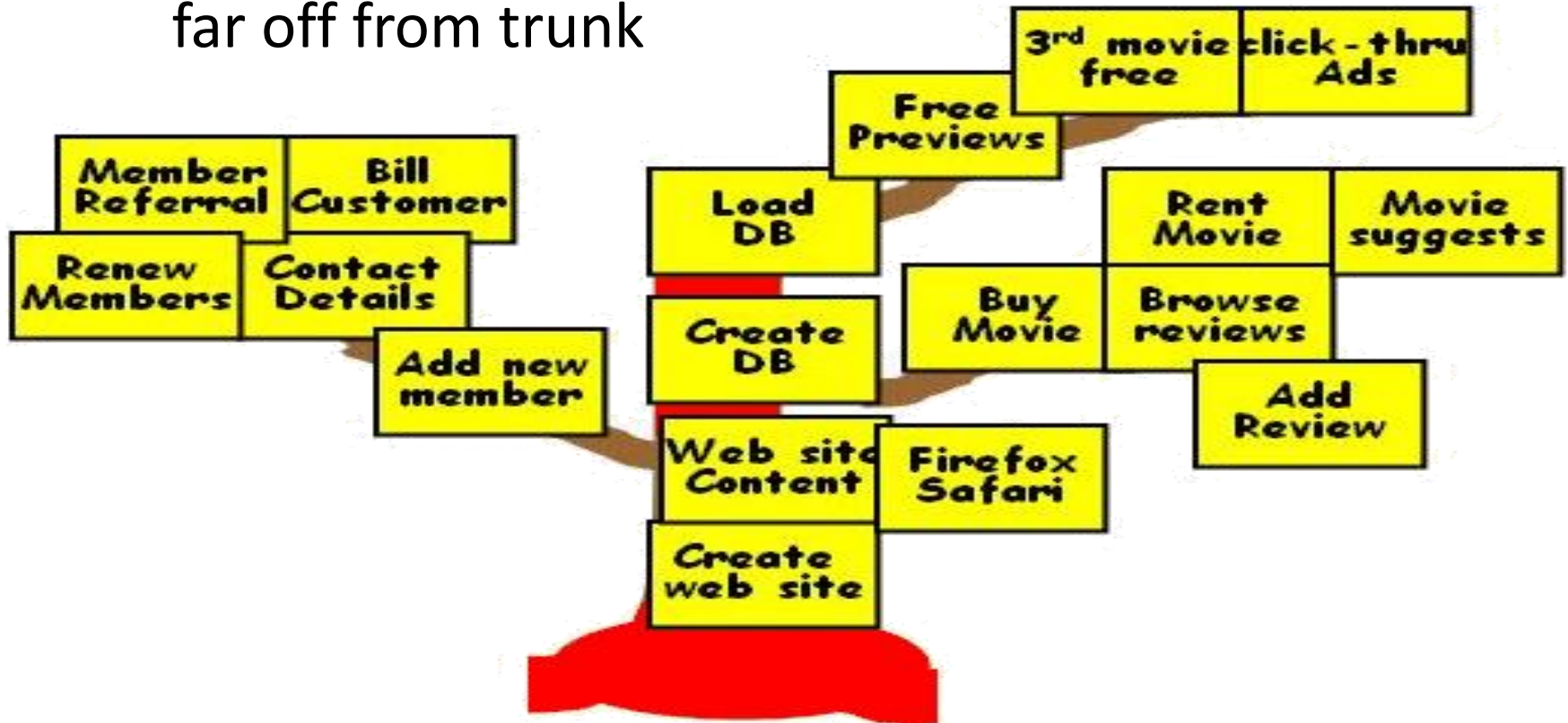
- Vision setting and requirements elicitation exercise
- Imagine post iteration / release period & answer “What made it successful?”
- Collect individual answers, remove duplicates, sort, rank
- Cognitive psychology: “People can recollect better than future prediction”!!



Agile games- Prune the product tree

To (brainstorm) gather and shape requirements

- Trunk represents infrastructure
- Related features in same branch
- Support features near trunk & dependent features far off from trunk

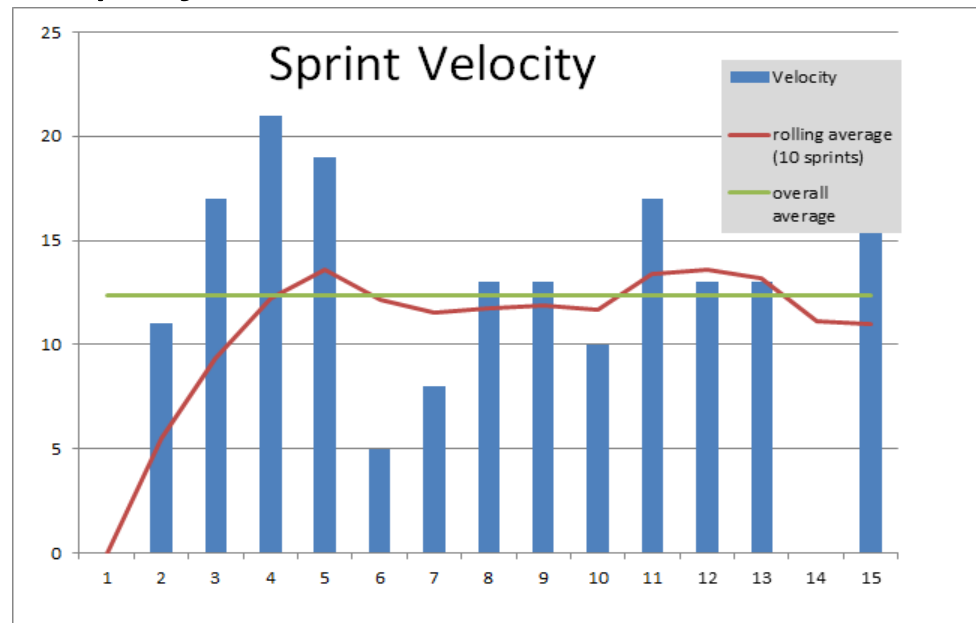


Risk based Spike

- Spike is a technical investigation to research an answer to a problem / for estimation
- Risk based spike is a short proof of concept (POC) that the team undertakes to remove uncertainty
- If we could not find a solution to the critical blocking problem, we reach a condition known as 'fast failure'
 - Fast failure helps to reduce losing time & money over projects bound to ultimately fail
- Spike helps learning enough about risky feature, technology, or process to better estimate, develop or fix an upcoming feature / defect

Velocity

- Story Points (or Ideal hours) successfully completed in a Sprint
 - Sum the **initial estimates** of 'done done' stories
 - Do not include bugs and rejected stories
 - Not a measure of productivity but proportionate to team composition
 - Not comparable across two projects
- For volatile team, calculate velocity per resource
- Client expects velocity improvement over time
- Velocity can be used to determine sprint scope, release cost estimation...



EXERCISE – AGILE PRACTICES

TASK BREAKDOWN TEMPLATE FOR A SAMPLE STORY

Story	A Planner can enter a new contact in the address book so that he can call or email the person in the future	
ID	Task	Estimate Man/Hrs
1	Identify tests to be performed	
2	Create contact-entry screen	
3	Revise database schema	
4	Implement persistence layer	
5	Implement the application logic	
6	Create unit and integration tests	
7	Create automated UI tests	
8	Write manual test cases	
9	Deploy to QA environment	
10	Execute manual test cases	
11	Execute smoke test	

EXERCISE – TASK BREAKDOWN & ESTIMATING

Use task-breakdown template to break the estimated stories to Tasks

Tasks should include all implementation, testing, and bug fixing work

Swap with neighbor, review, and revise

Time: 10 minutes

EXERCISE – TASK BREAKDOWN & ESTIMATING

1. Discuss each task, estimate it in person-hours
2. Put estimates on the template, by the tasks
3. Compute new Story estimate
 - a. Sum task estimates for each Story to provide a task total
 - b. Divide by 8 to get Person-Days
 - c. Update the backlog with revised Story estimate

Time: 10 minutes

9. TOOLS & TECHNIQUES

Progressive Elaboration

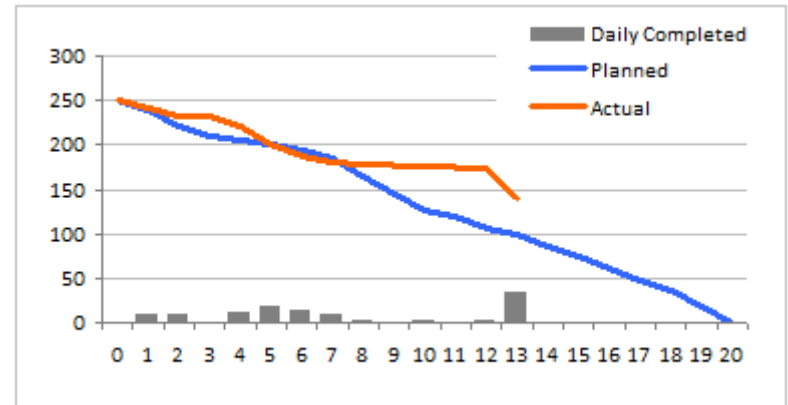
- Process of adding more detail as information emerges
- Plan, Requirements, Design, Risk...
- Rolling wave planning
 - Form of Progressive elaboration
 - Near future is expanded in detail
 - Distant future at overview level



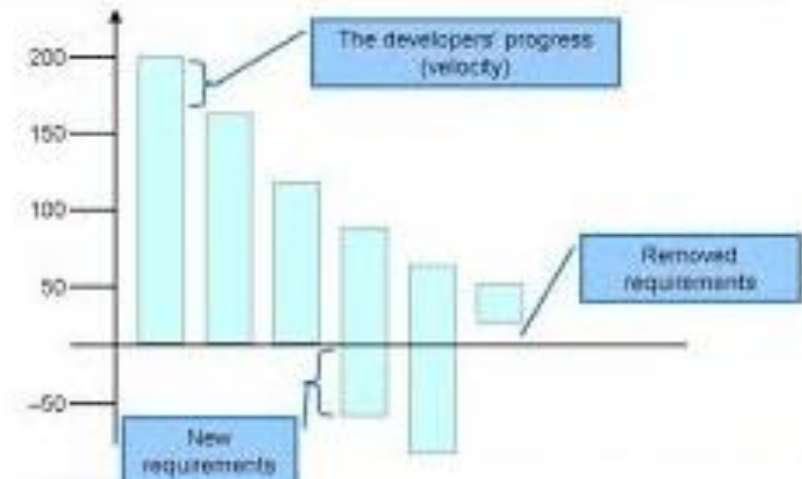
Burn charts/graph

- Visual Performance tracking tool
- Useful to gauge performance, take corrective actions, forecast, change scope...
- 3 types
 - Burn down chart – Work remaining Vs Time
 - Burn up chart – Work completed Vs Time
 - Combined chart - Both
- Can be applied at any level
 - Sprint, Release, Multiple release
- Chart type
 - Line, Bar
 - X axis: Time; Y axis – Work (story points, ideal time)

Burn down charts



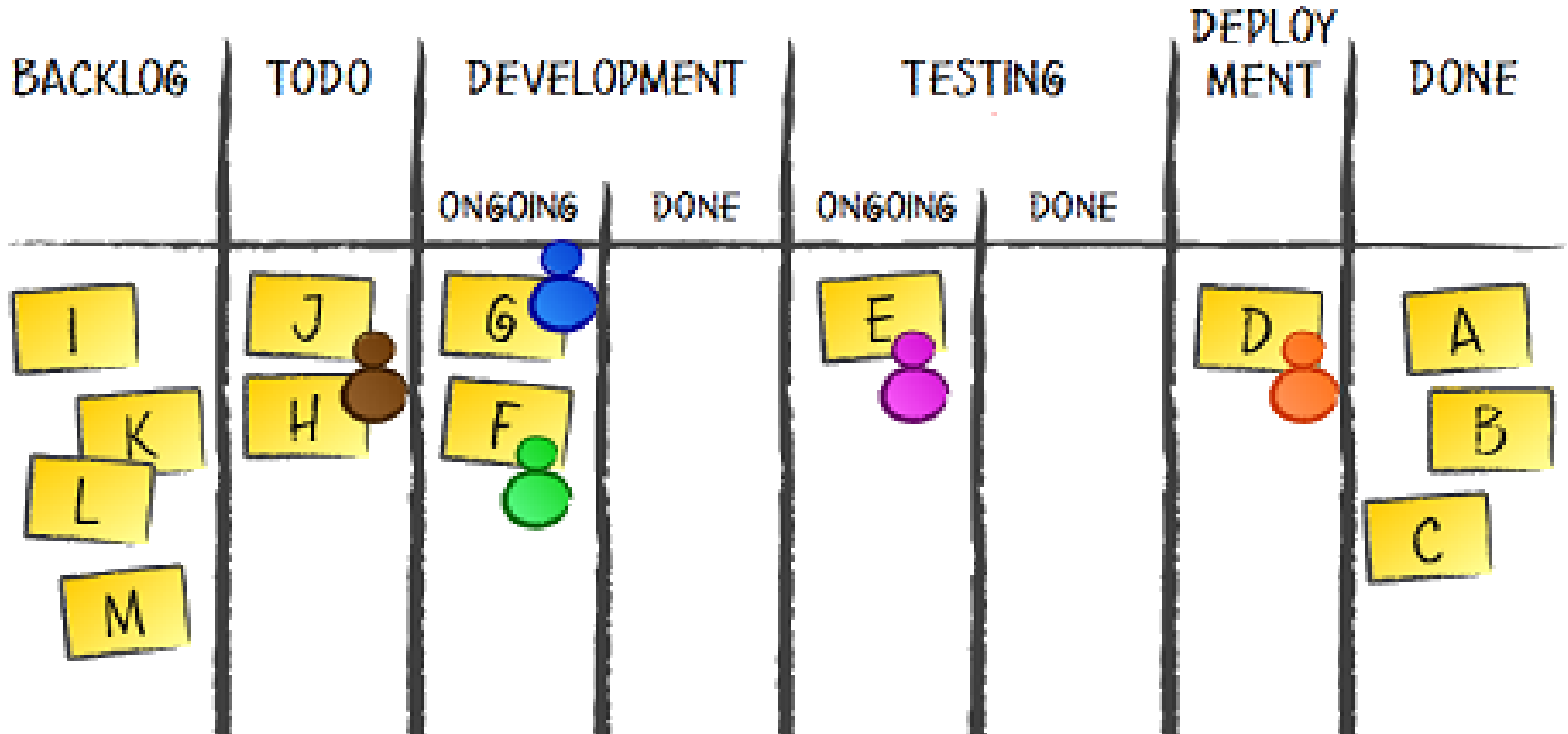
A more powerful release burndown



- Estimate to complete Chart
- It can include Planned/expected progress
- Does not show up any change in the total story points during sprint

- Length of the bar indicates amount of work left
- Movement of the top of the bar indicates amount of work done
- Movement of the bottom of the bar indicates changed work

Kanban / Task board



Kanban board - Types

- On time scale
 - Release kanban, Sprint kanban, and Task Kanban
- On task scale
 - Epic kanban, Feature kanban, Story kanban, Task Kanban
- Task Kanban
 - Shows flow of tasks. Useful to team for daily tracking

Stories backlog	Tasks queue	WIP	Tasks complete	Stories complete

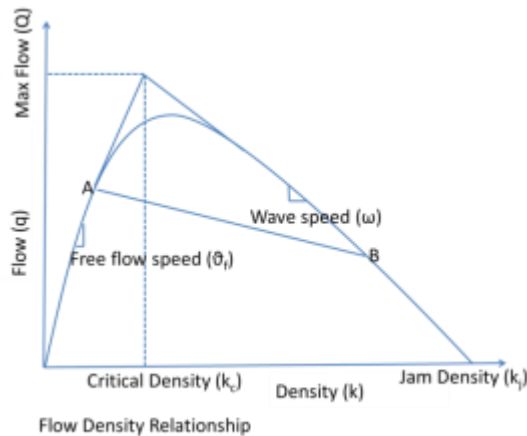
- Story Kanban
 - Shows flow of stories. Useful to PO, Customer, Business users...

Features backlog	Stories queue	WIP	Stories complete	Features complete

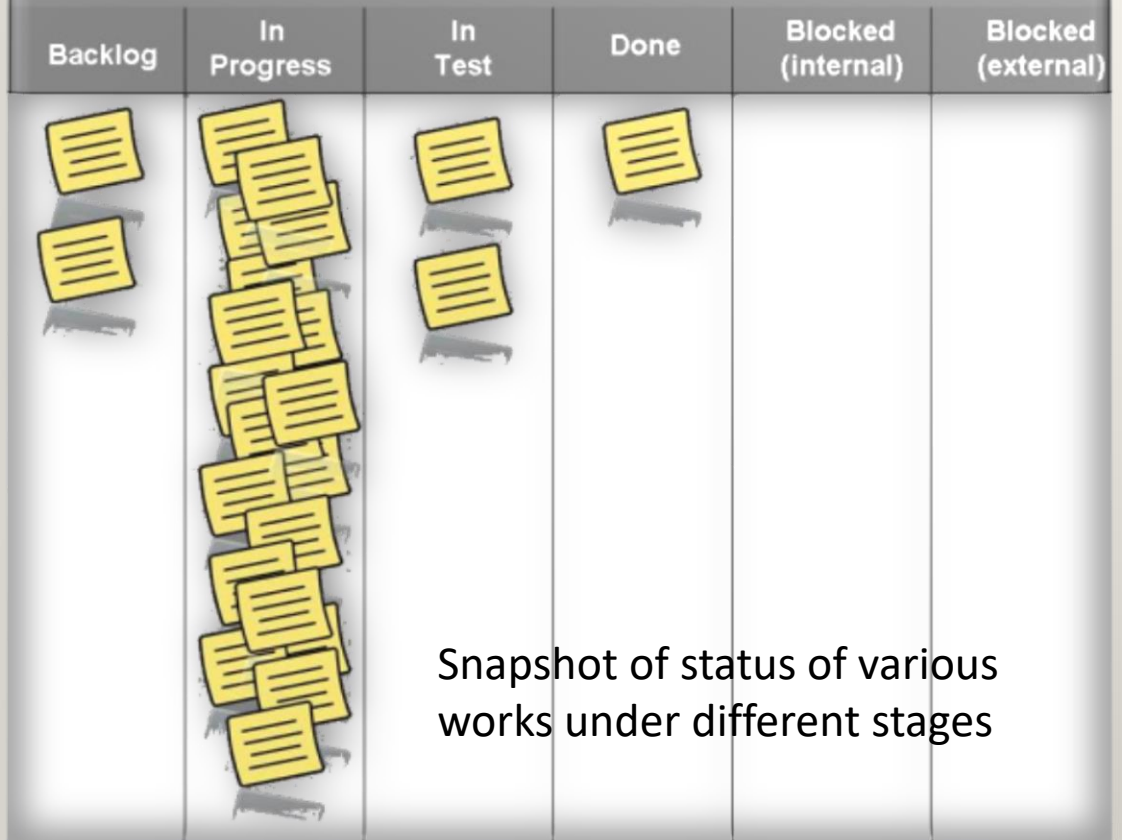
- Feature Kanban
 - Shows flow of features. Useful to Marketing department

Epics backlog	Features queue	WIP	Features complete	Epics complete

Limit WIP



Example of a first attempt at a Kanban board where too many things are "in progress"



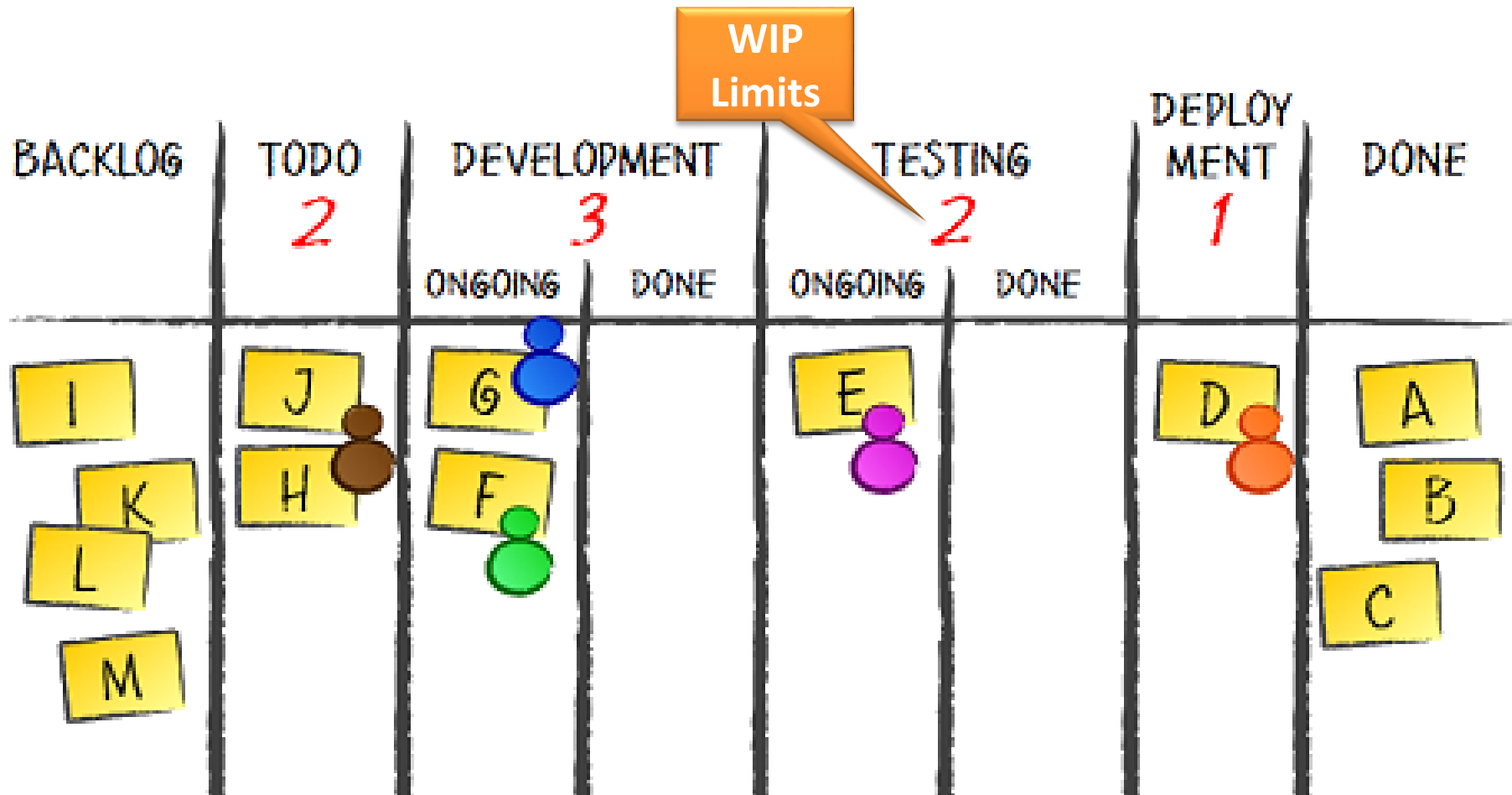
Optimize work throughput;
Not resource optimization

Limit WIP

- Optimize throughput, Reduce waste
- Powers Kanban board
 - To avoid over commit, inefficiency, bottleneck masks
 - Minimize waste (rework), investment capital, cost
 - Help Focus
- Little law: Cycle time $\propto$ Queue size
 - Cycle time is 'how long going to wait for benefits'
- Bottleneck
 - Inability of next step to consume work
 - Overtaking at that stage
 - Stop working on other streams to focus fully focus on bottleneck stage to clear it out, put a WIP limit on that stage to avoid future bottlenecks

Limit WIP

- Limit items in different WIP columns



Information Radiator

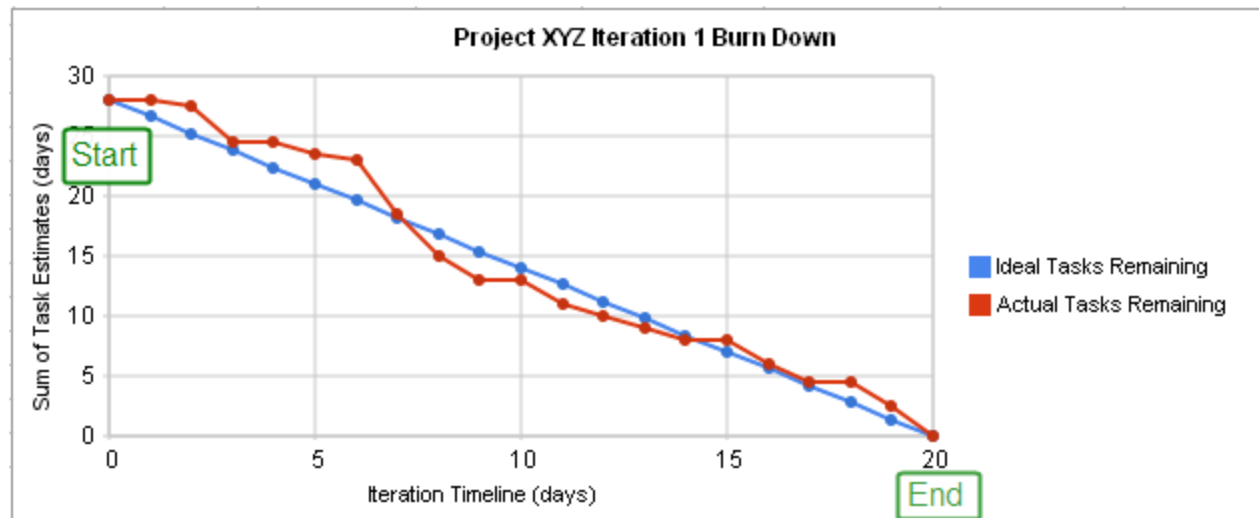
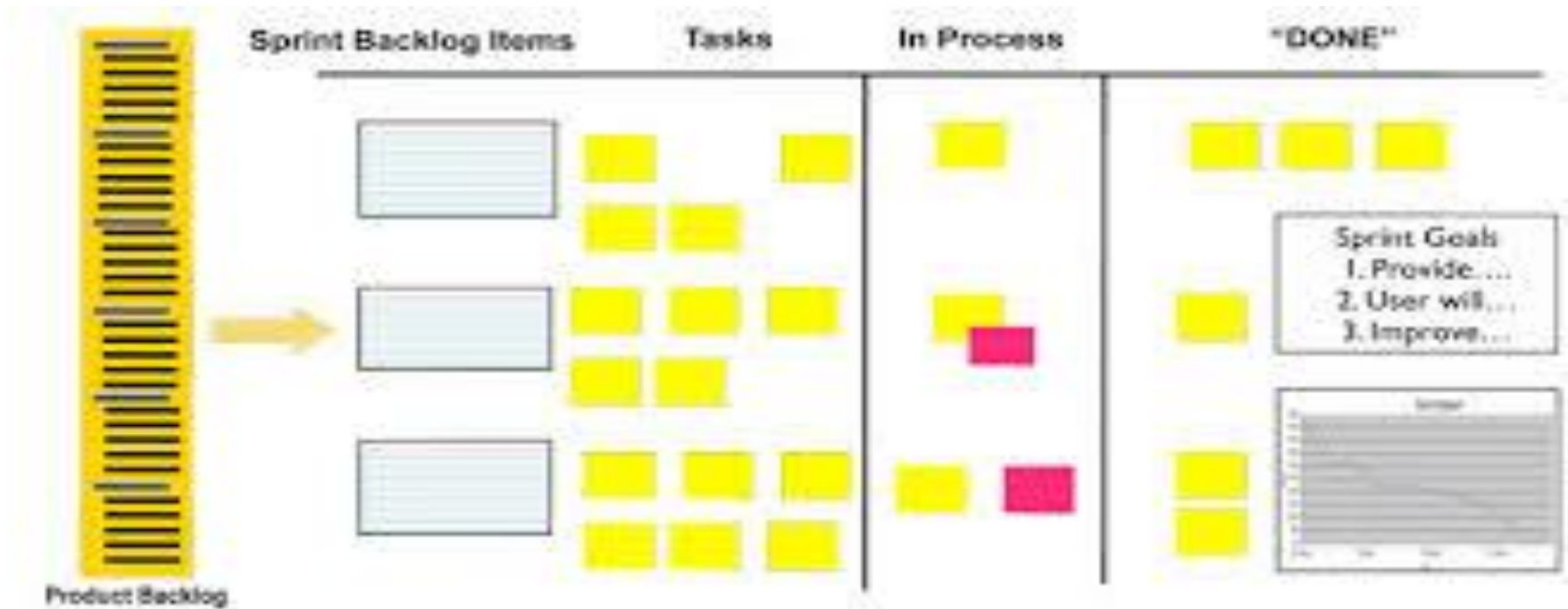
- Umbrella term for a number of highly visible ways to display info like Work Progress, Risks, Alerts, Project summary...
- Should be crisp, easy, stark, intuitive, maintainable, current, influential
- Whiteboards, flip charts, poster boards, Signal light or large electronic displays...
- Located in team traffic area

Information Radiator

Information include

1. Work items & their status, Impediments, Forecast
 - Task / Kanban boards, Parking lot graph, Burn up/down chart, EVM
2. Risks
 - Risk burn-down chart, Risk register
3. Health
 - Continuous integration build health indicators, retrospective findings
 - Velocity trend
 - Defect cycle time (Defect fix time–Defect seed time) trend
 - Open defects trend

Information Radiator: Samples



Feedback techniques for product

- Intangible, Unique product
- Gulf of Evaluation
 - Semantic gap in understanding



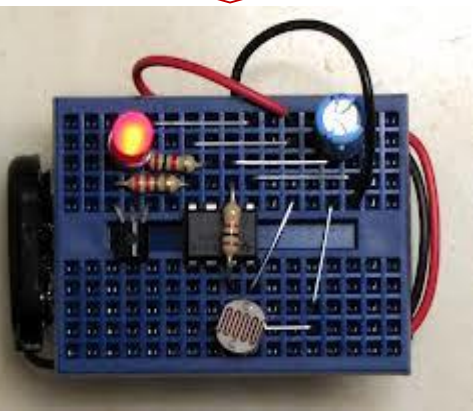
- Customer Confirmation / Feedback
 - Requirements evolve with prototypes, simulations & demonstrations
 - Customer valued (re) prioritization
 - Iteration demo:
 - IKIWISI (I Know It When I See It)
 - Aids Right Shaping, Innovation
 - Iteration review + Iteration Planning
 - PB grooming

Prototypes

Proof of concept (POC)

(bread board)

- Prove potential design



Visual prototype:

- Appearance, color, fonts, surface textures without functionality
- Reviews, Market research

Form study prototype

- Basic size, look & feel without functionality
- Prove Ergonomic factors



Functional / Working prototype:

- Simulates the final design, aesthetics, materials & functionality of the intended work as much as possible

Prototype in Greek means “Primitive form”

Wireframes

- Low fidelity rapid visual prototypes
 - Simple block diagrams that show the placement of elements in a user interface with annotation for functionality and flow between screens
 - Typically free from graphic design features like colors, typography or actual images
 - Easy to create and modify
- Try to answer the following questions:
 - What elements will be displayed in the UI?
 - How will the elements be organized?
 - How will the interface work?
 - How does the user interact with the application / website?
- Uses:
 - CONVERSATIONAL ICE BREAKER for requirements shape up
 - COMMUNICATION TOOL to agree on agile user stories

Wireframes

The wireframe shows a music store interface. At the top is a navigation bar with links: [Home](#) --> [Music](#) --> [Genre](#) --> Artist, and [Log out](#), [Your Cart](#), [Your account](#). The main content area is divided into three columns. The left column contains a search section (annotated with '1') with a search box containing 'music', a dropdown arrow, a text box for 'artist name', and a 'Go' button. Below this is a 'Related Artists' section (annotated with '2') with links for 'Artist 1' and 'Artist 2', and a 'Popular Tracks' section with links for 'Track 1' and 'Track 2'. The middle column features an 'ARTIST NAME' section with a description: 'This is a description about the artist. This will talk about their bio and short listing of their discography'. To the right of the description is a placeholder for 'Album Art' (annotated with '3'), represented by a dashed box with an 'X'. Below the description is a checkbox labeled 'send me updates on this artist'. The right column has a 'Choose an album:' section with a list of 'Album Title 1', 'Album Title 2', and 'Album Title 3', a scroll bar, and a dropdown arrow. Below the list are radio buttons for 'CD', 'Tape', and 'LP'. The price is listed as '\$16.99', and there is an 'Add to Cart' button.

- 1 For Q1 release, music search only
- 2 Related artists determined by user purchasing data mining
- 3 Album art to be approved by legal

Personas

- Archetypical (exemplary) user of a system, an example of the kind of person who would interact with the system
 - Fictitious people which are based on your knowledge of real users
 - Grounded in reality, goal-oriented, specific, and relevant
 - Create distinct types of end users
 - Helps to gain insight into end users requirements & Prioritization
 - Easy to Understand & follow
 - Personas are not a replacement for requirements
- Good persona is highly personalized
 - Give life to end users with real names, personalities, motivations, photo...

Personas - Sample

Angel, Small business owner

*Social Media Marketing
Key Decision Maker*



Demographics

- Age 32-39
- Skews Female
- \$90,000 / year
- Urban location
- Master's degree
- Married, no kids

Goals & Challenges

Save time online
Find interesting content to share
Maximize social media resources

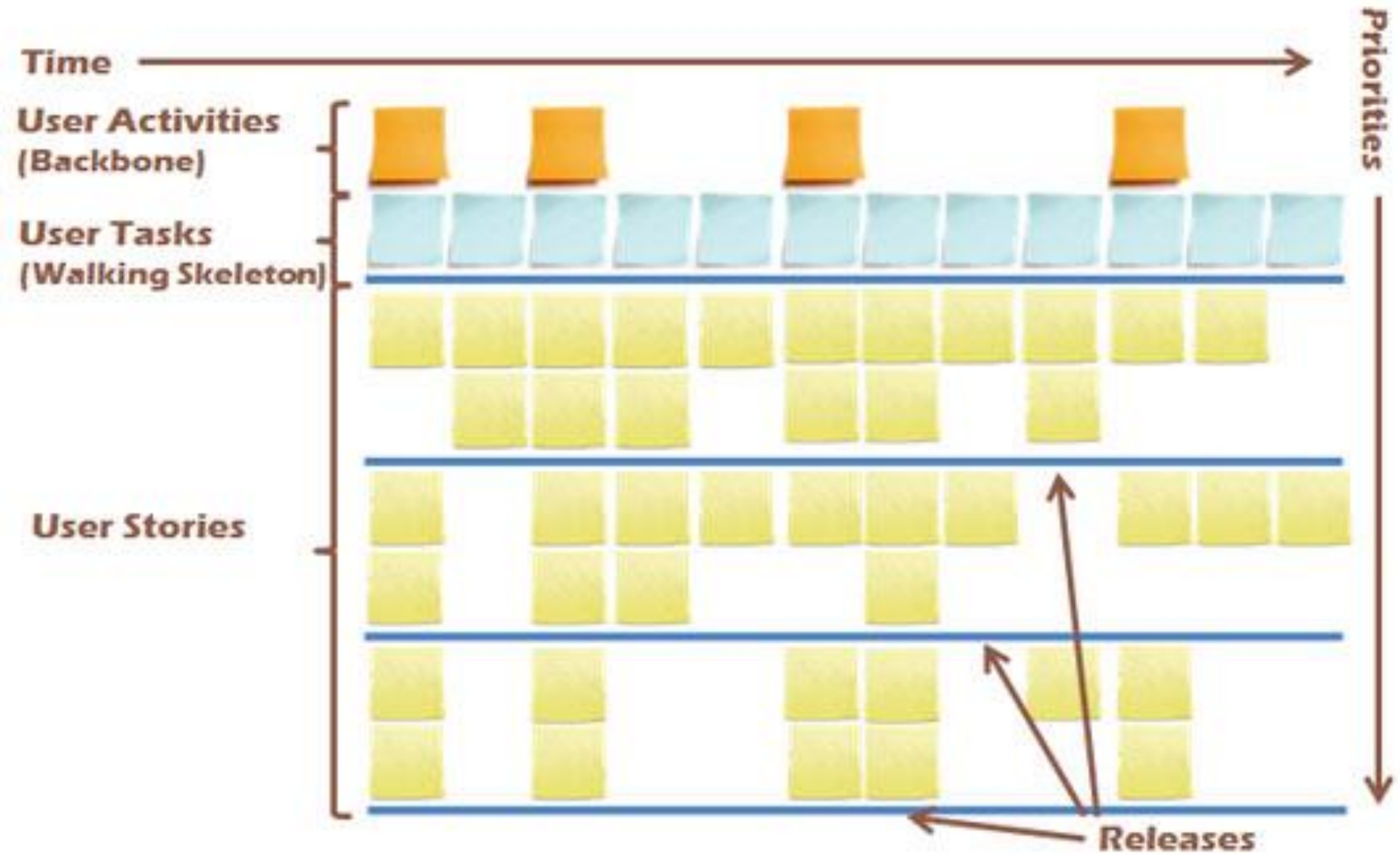
How we can help

Schedule posts to a queue
Content suggestions

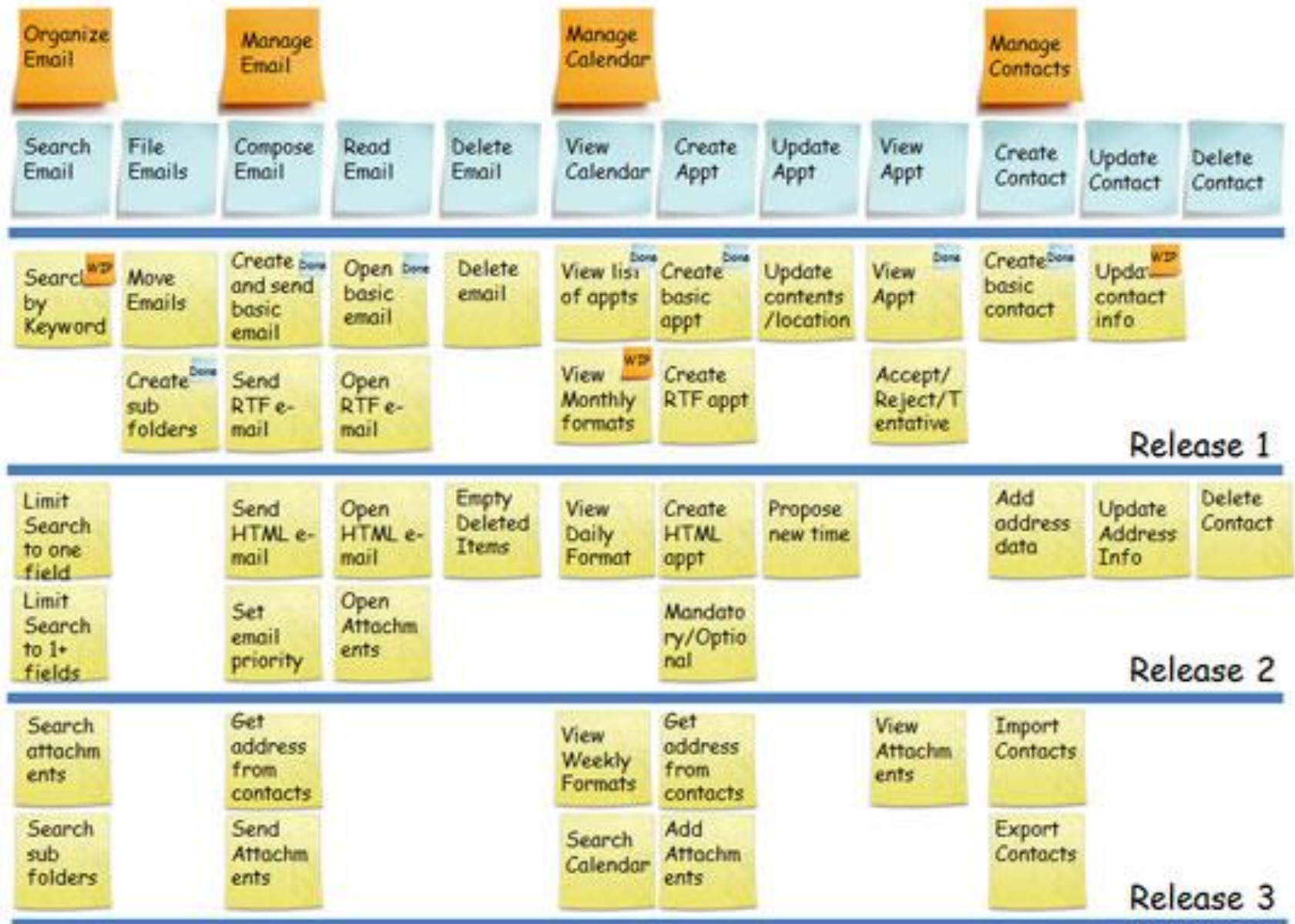
Story Map

- Features arranged horizontally based on workflow
- Features arranged vertically based on necessity / Sophistication
- Judicious horizontal swim-lanes to group features into releases
 - Guided by Product goals, MMF
- Benefit:
 - Arranges user stories into a useful readable model
 - Identify holes and omissions in your backlog
 - Help set priorities & plan holistic releases
 - Like a geo map, the specific location of a story gives you rich context

Story map: Parts



Story map - Example



Agile Tooling

- Disadvantages of Sophisticated tools:
 - Data accuracy perception increases
 - Creates barriers for seamless SH interaction
 - These tools disguise the volatile nature of what is being analyzed !!
- Agile uses a **low tech, high touch** tools
 - Information radiators
 - Kanban boards
 - Index cards for user stories
 - Planning poker

Distributed Agile tools

- Agile Project Mgmt System (APMS)
 - Provides overall status to entire team
 - Create & manage backlogs, Plans, iterations, releases & projects
 - Monitor & produce necessary metrics, charts & reports
 - Rally, VERSIONONE, Jira
- Visual card walls - Digital 'Information radiator'
- Smart boards: automatically captures design screens
- Digital camera: capture screens / white boards quickly
- Wiki, Portal...
- CASE tools: to extract design from code, DB tables
- Automated UT & Regression testing
- Continuous integration, build & release mgmt..
- Configuration mgmt..

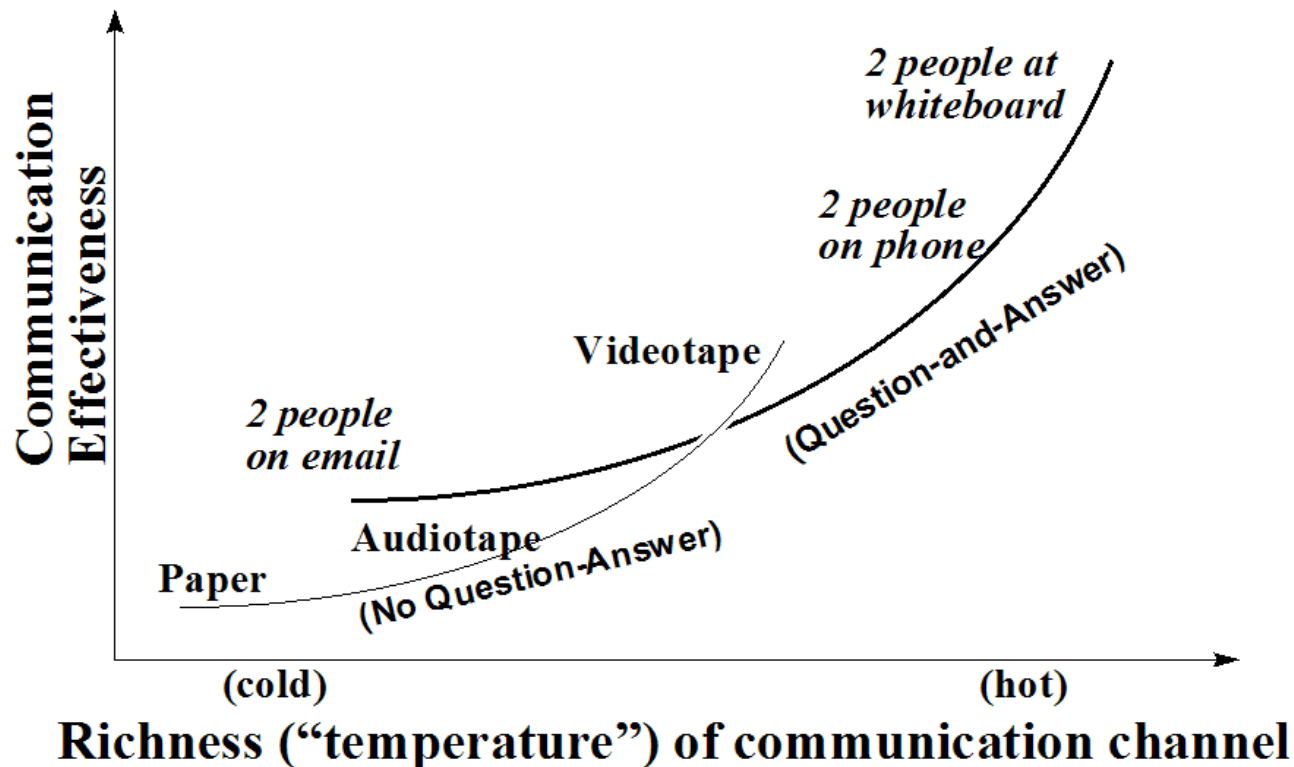
CASE = Computer Aided SW Engineering

BEING AGILE

10. COMMUNICATION IN AGILE

Communication Management

- Identify SH communication requirements
- Effective Communication Management
 - Info provided in right format, at right time with right impact on both sender & receiver



Communication

- Crucial in projects produce intangibles
- Gulf of Evaluation
 - Semantic gap in understanding
- Information Refrigerator
 - Have to open up and dig around in before you find the ketchup you're looking for
 - Example: Project Documents
- Agile promotes transparent & seamless comm.
- War room
 - The team meets in a single room for problem solving
 - Agile recommends the team to work continuously in the war room itself and call that room as 'information space'

Osmotic communication

- Adopted from Crystal methodology
- Cost effective, Tacit Knowledge sharing
- Indirect information transfer through overhearing conversations or simply noticing things happening around you
- Commons & caves
 - Commons is the group work space whereas caves (private room) used for personal things OR for think-tank
- Co-location: Simple & Effective
- Virtual team
 - Simulate osmosis comm. via Live video conf., group chat IM, Wiki, Emails CC to team, SharePoint and other collaboration tools

11. STAKEHOLDER (SH) MANAGEMENT

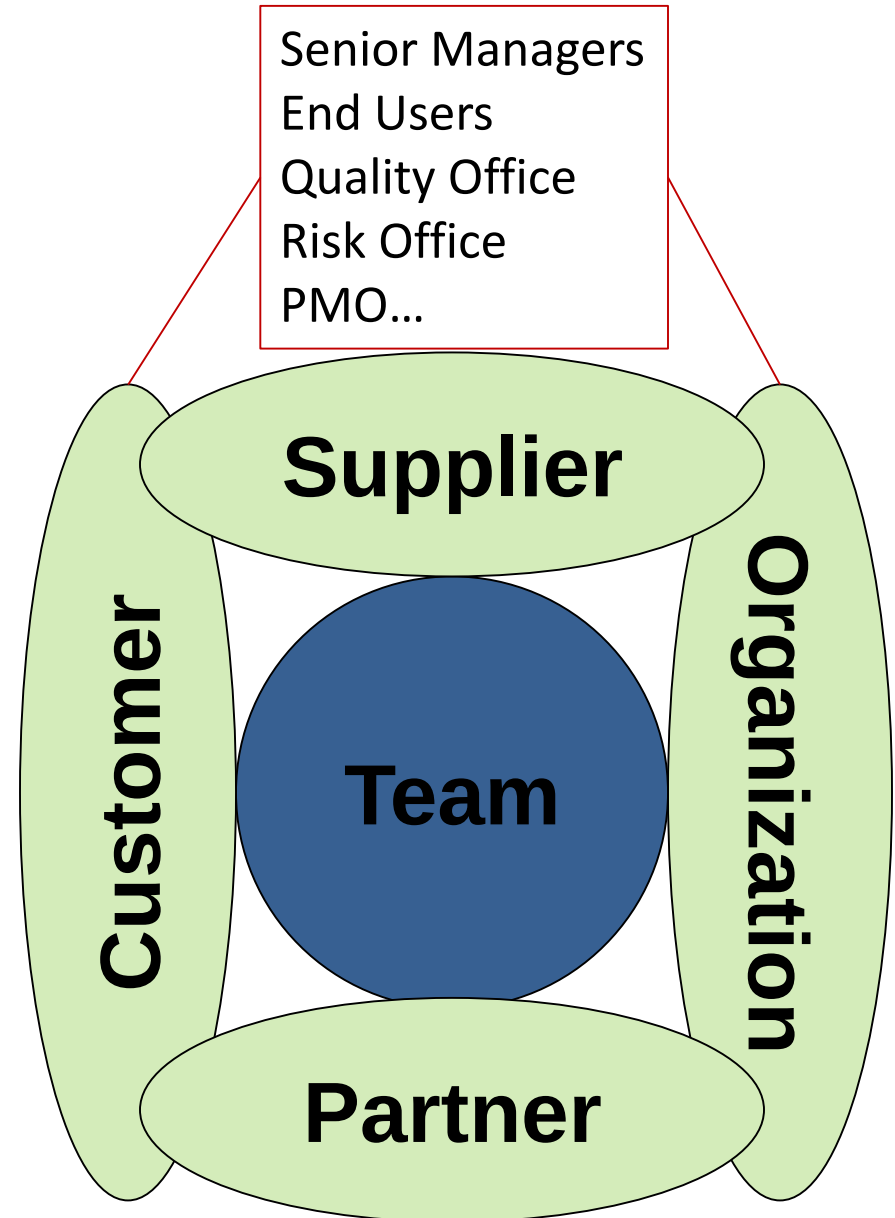
Project Stakeholders

“Individual, groups, or organization that could impact* or be impacted* by a decision, activity or outcome of the Project”

*Positively OR Negatively

Author Rob Thomsett proposes 3 levels of participants (stakeholders)

1. Critical
2. Essential
3. Non-essential



Stakeholder Engagement: Why?

- Intangible output
- 20-50% requirement changes
 - Fast changing technology & user preferences
 - Customer requirement evolves
- Gulf of Evaluation
 - Customer: “Herbivorous, 1 trunk, 4 legs, black color, should carry load”
 - Team: “also gives milk!!”



Stakeholder Management



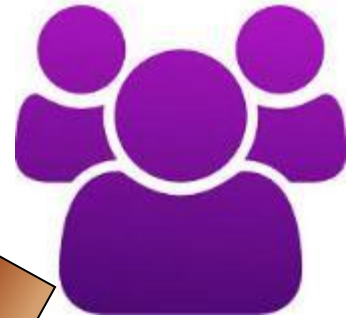
- Identify right stakeholders
- Understand their expectations
- Educate them about agile, address their concerns
- Continually engage them in the project
- Show Progress & Capabilities
 - Frequently discuss what 'done' looks like
- Elicit changes early through multiple touch points like iteration review
- Establish a process for escalation
- The SHs who create impediments, use EI to address, and if not successful try to shield team from their disruptive or corrosive influence

*Creating a shared **Definition of Done** is crucial for managing SH expectations*

Incorporating SH Values



Project Vision



SH Priority

1. (Re) Prioritization
2. SH in retrospectives & plan meetings

Aligned Priority

Vendor Management

- Decide on buy items
- Prepare SoW & procurement documents
- Elicit proposals from potential sellers
- Evaluate proposals & award contract
 - If contract is agile, it must be flexible to allow changes and early & frequent deliveries
- Monitor & Control Item development
- Receive-Integrate



SOFT SKILLS

Soft Skills

- Soft Stuff is the hard; Hard stuff is easy !!
- IQ peaks in 20s; EQ continues to develop
- Critical Soft ~~talents~~ Skills
 - Negotiation
 - Active Listening
 - Facilitation Skills
 - Conflict resolution
 - Globalization, culture, team diversity
 - Collaboration in Distributed teams
 - Emotional Intelligence (EI)

Negotiation

- Discussion intended to produce agreement
- Richard Shell's process of negotiation in 4 stages

Understand “What you want”

Preparation

- Establish rapport
- Understand others
- Set common goals

**Exchanging
Information**

Bargaining

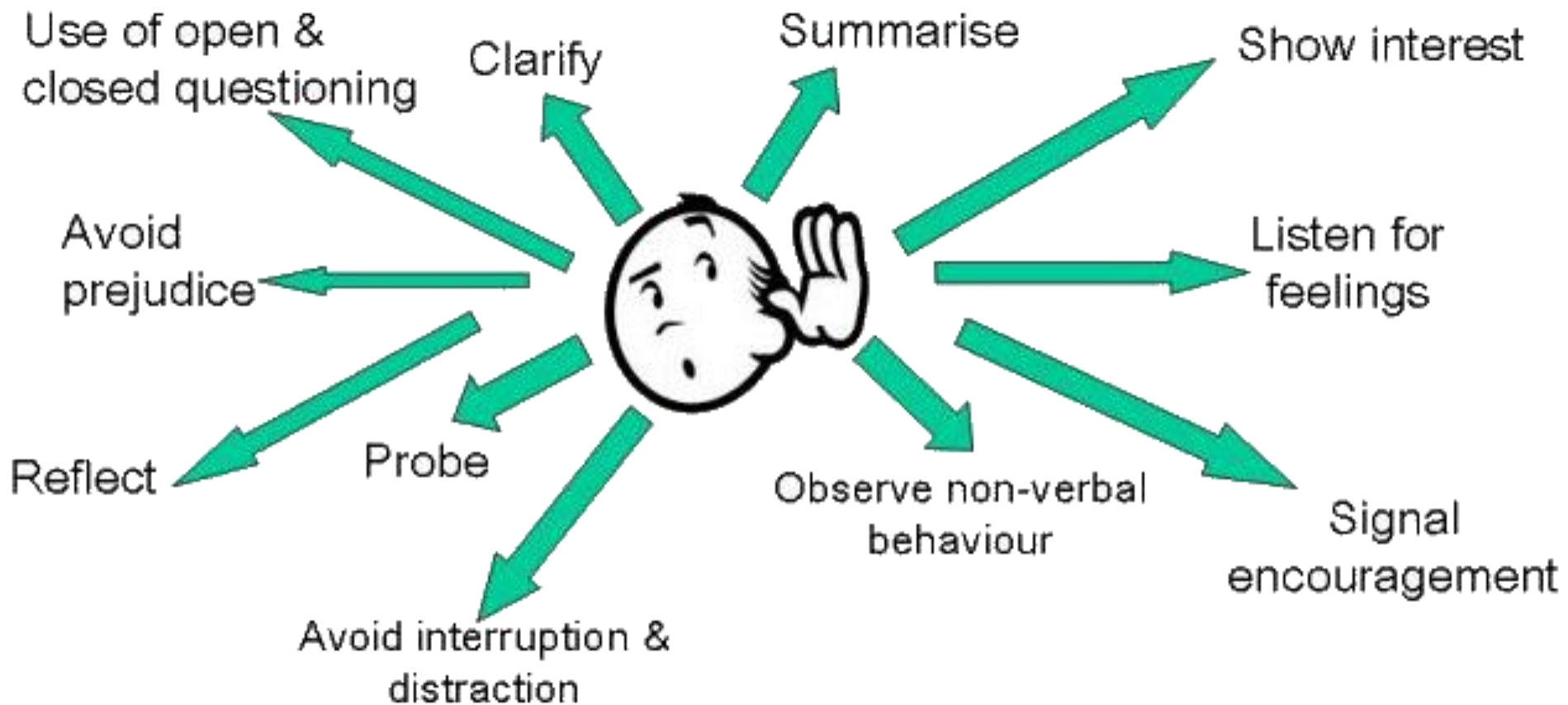
- Separate people from the problem
- Focus on interests, not positions
- Invent options for **Win-Win**
- Use objective criteria
- Analyze outcomes & risks

**Closing &
Commitment**

Summarize & Record agreements
& next steps

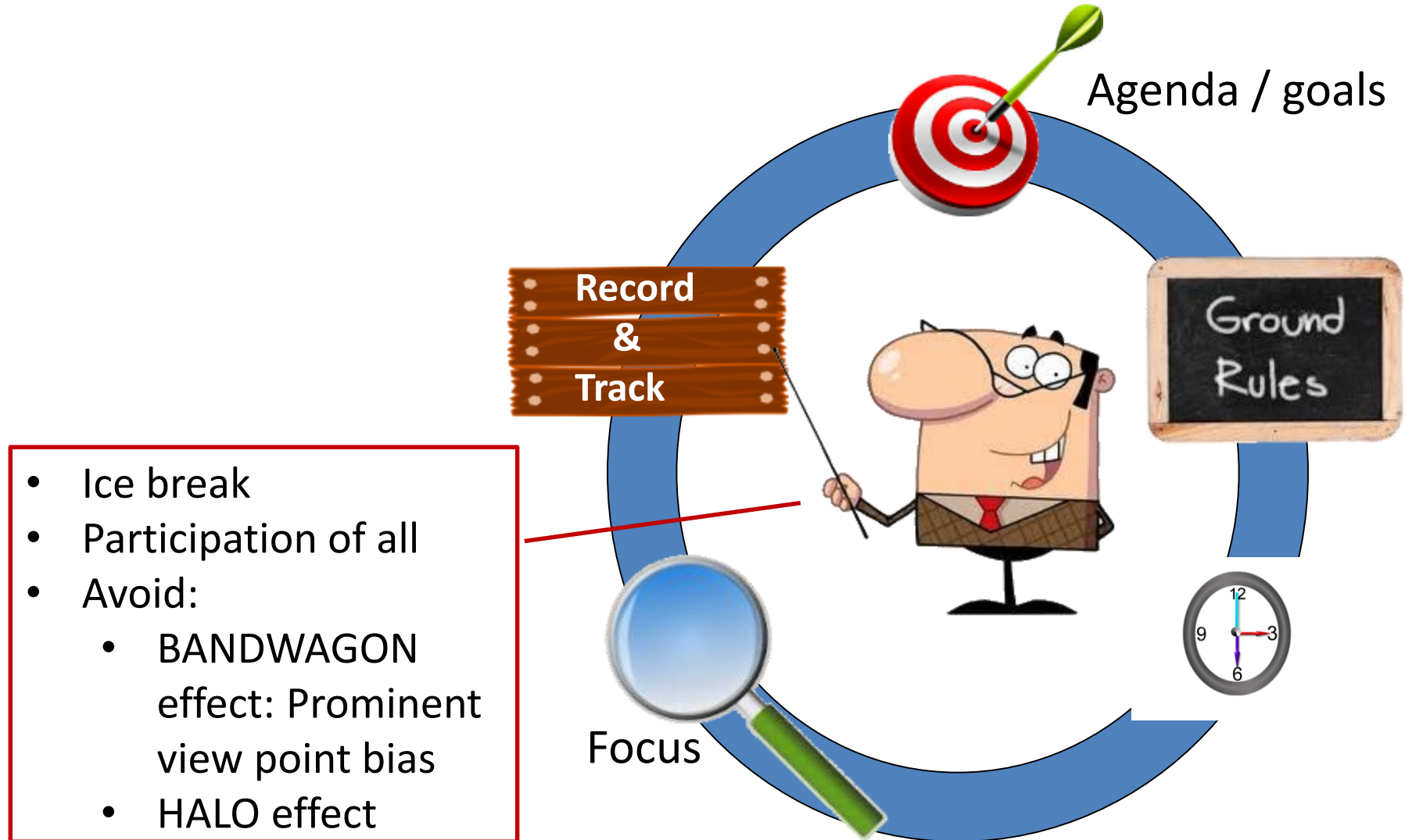
Active Listening

- Its about emphasize & seek to understand others; not necessarily agree with them



Facilitation Skills

Its about effectively facilitating meetings / sessions

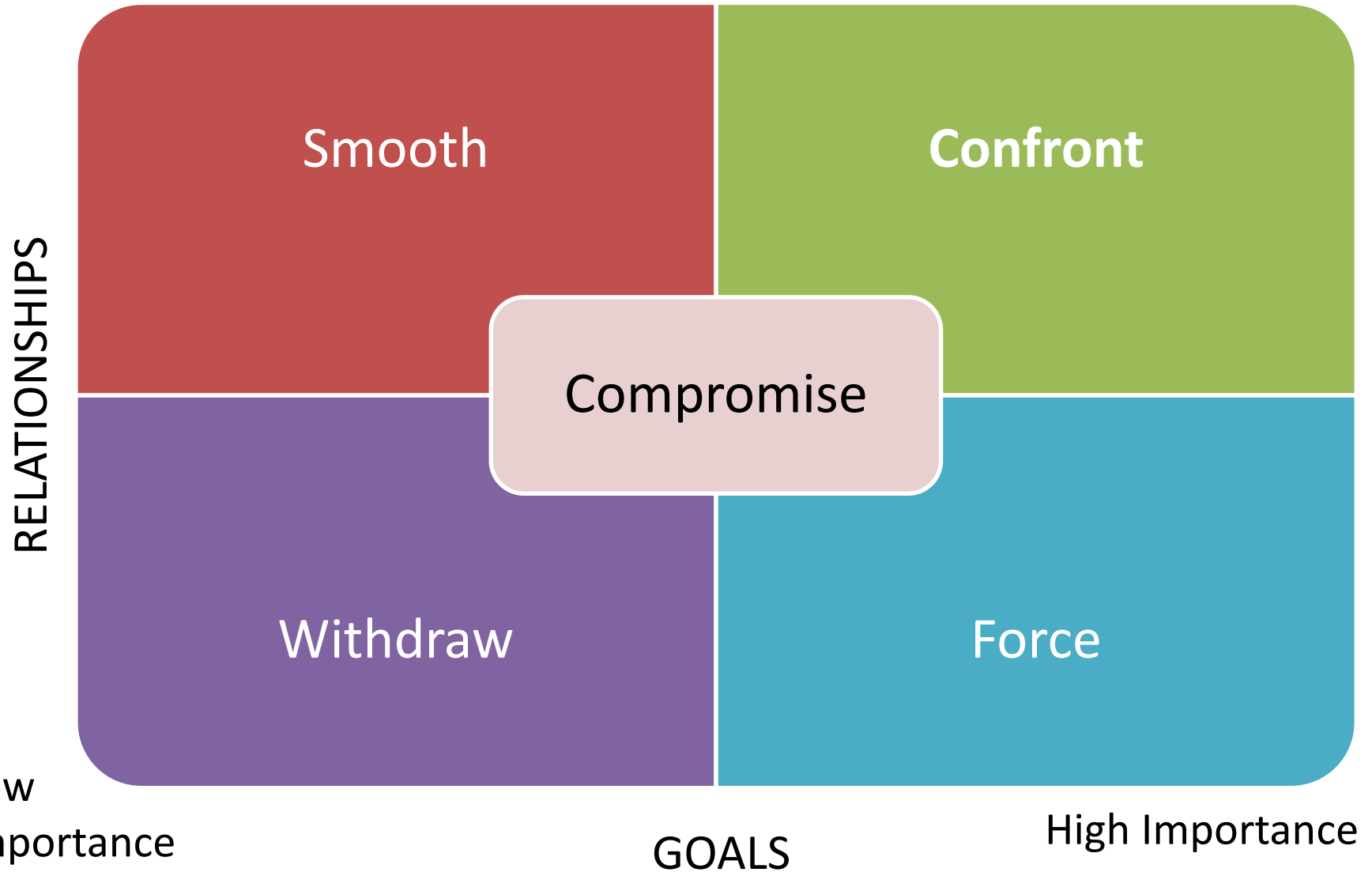


Conflict Management

- Conflict is inevitable in a project
- Sources of conflict include scarce resources, scheduling priorities, personal work styles...
- Prevention measures: Team ground rules, group norms and solid project management practices (like comm., planning, role definition)
- Conflicts should be addressed EARLY & in PRIVATE using DIRECT, COLLABORATIVE approach
- Successful conflict mgmt.. results in creativity, better decisions, greater productivity and positive working relationships
- Unresolved conflict can hurt the project

Conflict Resolution approaches

High Importance



Conflict Resolution approaches

- Collaborate / Problem Solving: confront the problem and NOT the person. Resolve the root causes of the conflict. Long lasting Win-Win solution. Favored method by PMI
- Compromise / Reconcile: Both parties sacrifice something for the sake of reaching an agreement. Lose-lose. Temporary / Partial resolving of the problem
- Forcing / Direct: using power/force for a solution; never solves the problem and not a good long-term solution. Win-lose / Lose-Lose solution
- Smoothing / Accommodate: Downplays conflict area and emphasize areas of agreement; diverting attention to GOODIES. Does not attempt a solution; only tries to diminish the problem
- Withdrawal / Avoid: avoid facing the problem hoping the problem would vanish with time OR someone else will address to it OR 'do not care'

Globalization, Culture, Team diversity

- Virtual team
 - + Wide pool to choose from
 - Communication, Culture, Coordination challenges
 - Ex: Westerners Vs. Non-Westerners culture
- Initial Collocation sprints
- Temporary re-location
- Cross cultural trainings
- Induction training
- Maintain core team....

Culture shock is the disoriented feeling that people might experience when working in a foreign country

Collaboration in Distributed teams

- Collaboration → Synergic output
- Colocation offers F2F, Osmotic communication, Improved relations; which needs substitutes in virtual team

Team:

- Cross functional team at each site
- Clear role definition
- Initial Colocation Iterations
- Minimize Team churn
- Scrum of Scrums

Process:

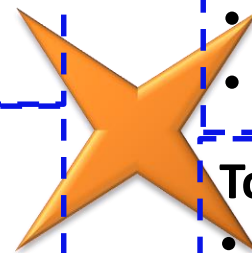
- Structured processes
 - Use online agile tools like issue analysis, product backlog
- Retrospectives: Effective virtual team
- Overlapping work / meeting hours

Communication:

- F2F
- Video conference
- Web collaboration tools
- Audio conference
- Chat
- VoIP
- Portals
- E-mail
- Interactive white-boards
- Ad-hoc communication
- More meetings

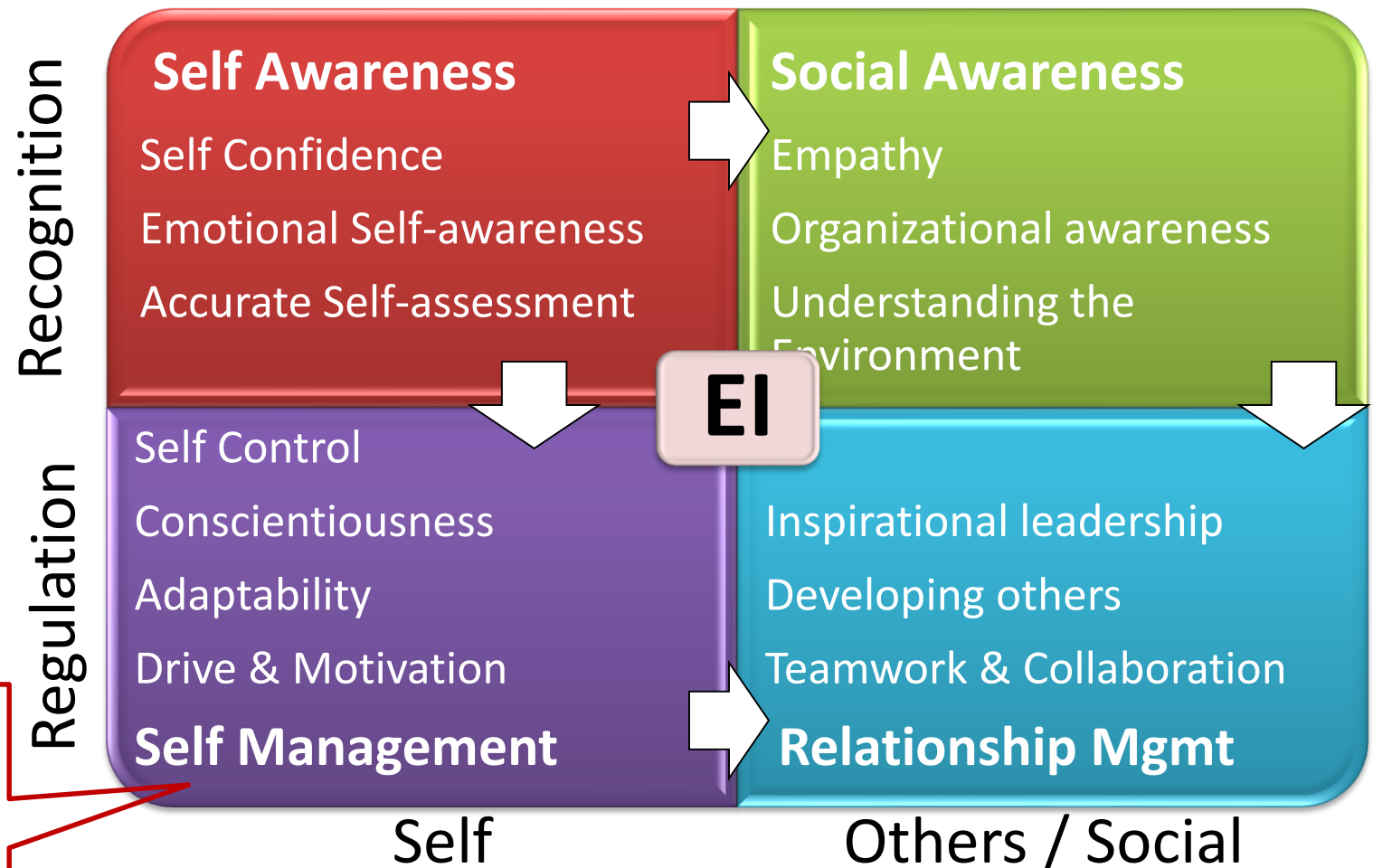
Tools:

- Information radiator
online substitutes
- Continuous integration



Emotional Intelligence – 4 Aspects

- EI, measured as an Emotional Intelligence Quotient (EQ), is a measure that describes the ability to identify, assess, and manage the emotions of one's self, of others, and of groups
- The leader's mood casts on team



PMI's Code of Ethics & Professional Conduct

- 4 Values: Responsibility, Respect, Fairness, Honesty
 - Mandatory Standards: MUST to follow
 - Aspirational Standards: Goal (not optional)
- Commitment to ethical, professional conduct and obligation to company/Government regulations
- PM should be a leader. Do
 - Deal with issues in direct manner
 - Act fairly, ethically & legally
 - Be Open & up-front
- Applicable to PMI members, Certificate aspirants / holders
- To build confidence, credibility & reputation for project management profession

PMI's Code of Ethics & Professional Conduct

1. Responsibility: Is our duty to take ownership for the decisions we make or fail to make, the actions we take or fail to take, and the consequence that result
2. Respect: Is our duty to show a high regard for ourselves, others and the resources entrusted to us. Resources entrusted to us may include people, money, reputation, the safety of others, and natural or environment resources
3. Fairness: is our duty to make decisions and act impartially and objectively. Our conduct must be free from competing self interest, prejudice, and favoritism
4. Honesty: Is our duty to understand the truth & act in a truthful manner both in our communication and in our conduct

13. HIGH PERFORMANCE TEAM

Team motivation

- Motivation levels:
 - Resisting → Passive → Active → Committed → Passionate Innovation
- Motivational factors:
 - Maslow's hierarchy of needs
 - Herzberg's two factor theory
 - Big picture & Importance of the project
 - Clear roles and responsibilities
 - Align team members' aspiration to project goal
 - Recognition



Maslow's hierarchy of needs

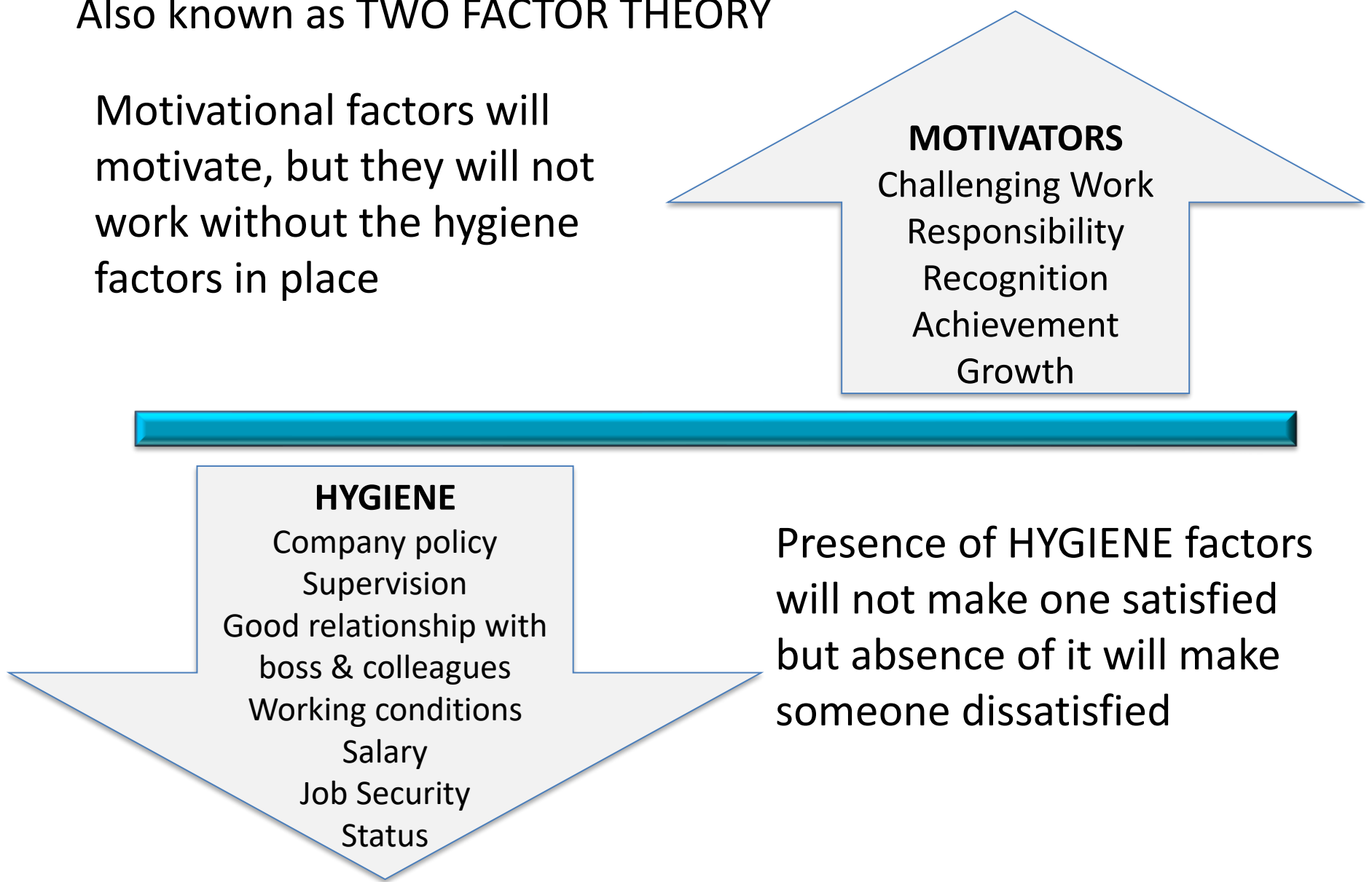
Understanding the level of team members helps to understand & work with them better



Herzberg's Motivation Hygiene theory

Also known as TWO FACTOR THEORY

Motivational factors will motivate, but they will not work without the hygiene factors in place



MOTIVATORS
Challenging Work
Responsibility
Recognition
Achievement
Growth

HYGIENE
Company policy
Supervision
Good relationship with
boss & colleagues
Working conditions
Salary
Job Security
Status

Presence of HYGIENE factors will not make one satisfied but absence of it will make someone dissatisfied

Adaptive Leadership

Tuckman team formation +

Blanchard & Hersey's styles of Management



Building high performance team

Teams that are highly focused on their goals and achieve superior business results consistently

The leaders manage the principles and the principles manage the team

MANAGEMENT

- Autonomous
- Participative leadership
- Self Managing
- Self Policing
- Self Improvement

ENVIRONMENT

- Trust
- Open & Clear Comm
- Collaboration
- Good Conflict Mgmt
- Live in Constructive disagreement

- Sustainable pace
- Consistent high velocity

RESULT

PEOPLE

- Right People,
- Clear Roles & responsibilities
- Clear goals

TEAM

- Small
- Cross functional
- Common goals
- Committed, accountable & empowered team

High
Performance
Team

Building empowered team

- Empowerment
 - is about taking responsibility & ownership with adequate power
 - working independently towards common objectives
- Qualities:
 - Self organizing (the work)
 - Leverage on team experience & mgmt.. skills
 - Motivation & Commitment
 - Servant leadership
 - Self directing
 - Create team norms & take own decisions
 - Improve in Iteration retrospectives

Co-located teams

- Team seated in single location; Ideal for Agile
- Seamless F2F communication & visibility
 - ‘Caves & Common’ for privacy & think-tank
- Aid Osmosis communication



Team space

- Should facilitate Seamless communication & visibility
- Hygiene environment
 - Lighting, air, food, tidy, group seating, furniture, window...
- Minimal distractions
 - Space should not be noisy or disruptive to others
 - Commons & caves. Commons is the group work space whereas caves (private room) for personal things / quiet think
- Big visible information radiators
 - Kanban boards
 - Spring backlog
 - Product backlog
 - Whiteboard for discussions

Coaching and mentoring

Coach	Mentor
Task oriented	Relationship oriented
Short term	Long term
Performance driven	Development driven
• Develops specific skills for the task • May not be expert, but knows tricks of the trade	• Has a deep personal interest for long term development • Expert/Veteran to Junior
Sports coach	Master

Agile Coach



Shu



Ha



Ri



Teaching



Coaching



Advising

Self

- Ability to read room
- Cultivate curiosity
- Thirst for learning
- They risk being wrong

People

- Are good
- They can do
- Care them over products
- Work with team not with plan

Native
Wiring

Process

- Low tolerance for institutional reasons that hold people back from excellence
- Disequilibrium is essential. Chaos and destruction are simply building blocks for something better. Messiness is expected

Participatory decision models

- Self Organizing Team & Decision models
- DECISION FRAMING focuses on who gets involved in the decision process

Simple Voting



NO



Fist of Five voting

Thumbs up / down / sideways



Jim Highsmith's decision spectrum



Group Decision making



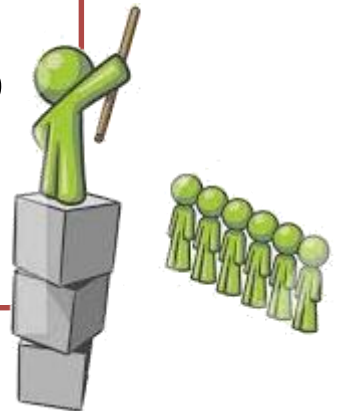
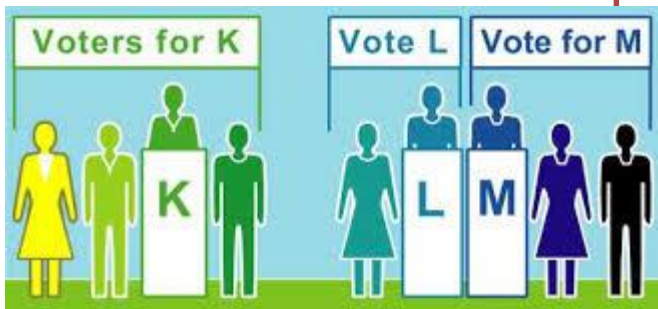
Unanimous
(100%)

Majority
($> 50\%$)



Plurality
(Largest Block)

Dictatorship



Management Vs Leadership

- Warren Bennis:
 - "Management is getting people to do what needs to be done"
 - "Leadership is getting people to **want to** do what needs to be done"
- Both are important

Management	Leadership
Tasks/things	People
Control	Empowerment
Efficiency	Effectiveness
Doing things right	Doing right things
Speed	Direction
Practices	Principles
Command	Communication

Leadership tools & techniques

- Model desired behavior
 - Honesty, Forward looking, Competent & Inspiring
- (Re)Communicate project vision
 - Metaphors, Mantras, project tweet, Elevator statement
- Enabling others to act
 - Foster collaboration by building trust & safe environment and sharing power
 - Exclusive tools → Inclusive tools
- Challenge the status quo
 - Let Improvement & Innovations thrives

Servant leadership

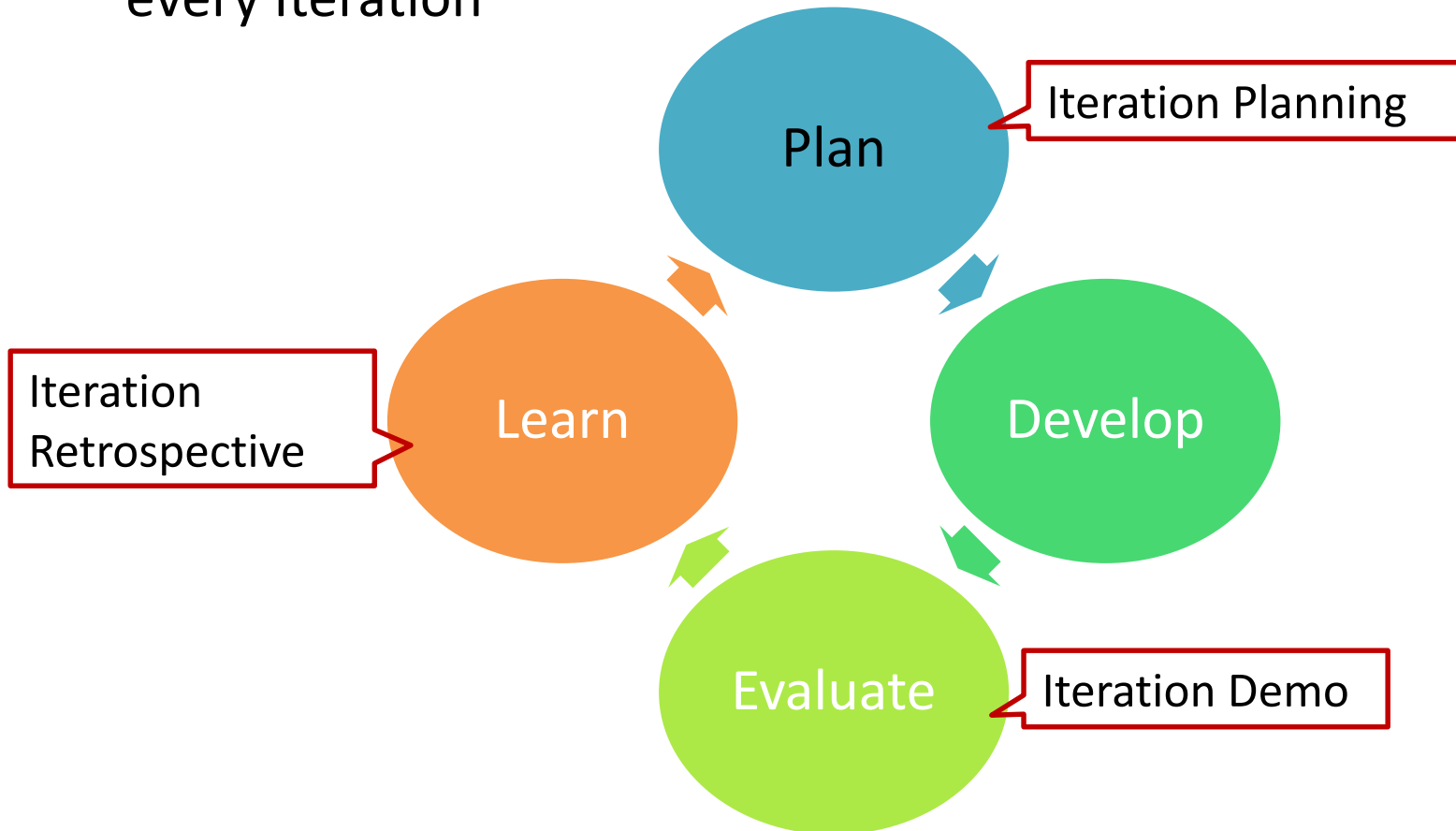
- Robert K Greenleaf
- Philosophy & Practice of leadership; Popular in agile
- Humble stewards of their organization's resources: human, financial, physical
- Build, support & leverage Self organizing team
- Emphasis on collaboration, trust, empathy, ethical use of power
- Duties
 1. Shield the team from interruptions
 2. Remove impediments to progress (log, daily standup meetings)
 3. (Re)Communicate project vision
 4. Carry food & water: Support, tools, compensation, encouragement, rewards & recognition



14. CONTINUOUS IMPROVEMENT

Continuous Improvement

- Ongoing process of enhancing the project approach and the product
- Agile lifecycle employs PLAN-DEVELOP-EVALUATE-LEARN for every Iteration

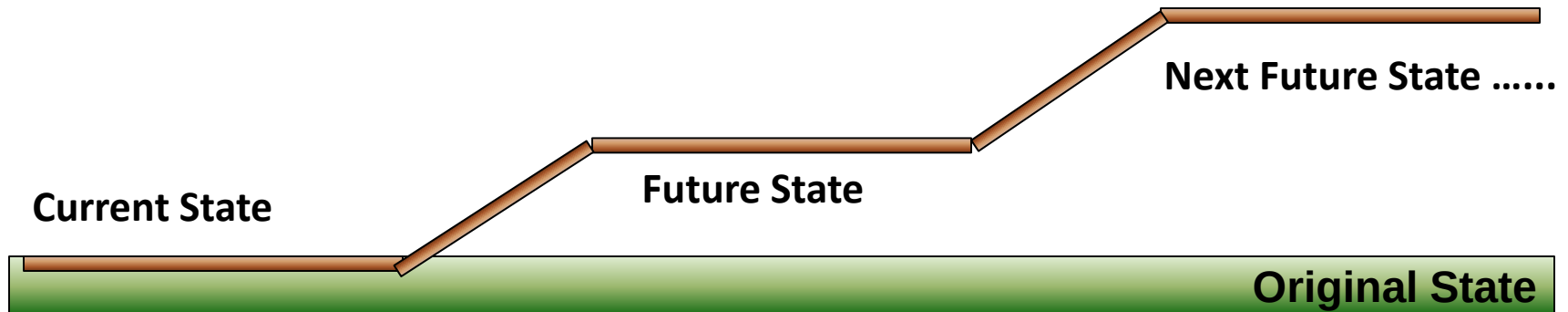


Continuous Improvement

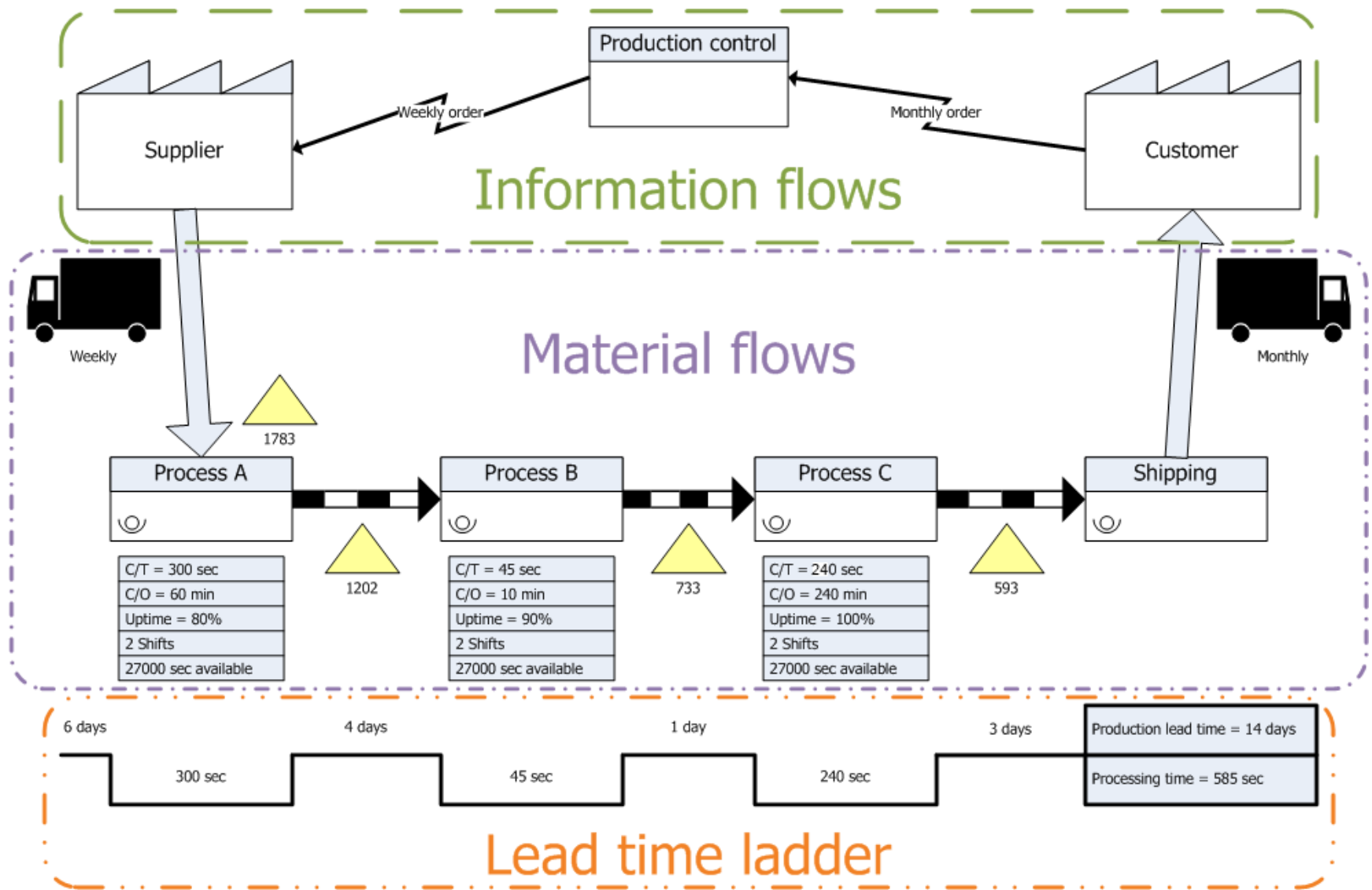
- Process:
 - Tools include VSM, Cycle time, Quality, Escaped defects, Variance & Trend analysis
- Product:
 - Tools include brain storming, Iteration demo
- People:
 - Tools include Failure modes and alternatives, knowledge sharing, Self assessment
- General:
 - Tools include Problem detection & resolution

Value Stream Mapping - VSM

- A Value Stream is the sequence of steps from concept to cash
- It includes the whole sequence – concept or customer order through delivery of value and/or receipt of cash
- It contains the people who execute, the systems, and the flow of information as well as materials



VSM - Sample



Quality Standards: Basics

- Conformance to specifications; Fitness to use
- Customer (External) quality & Technical (Internal) quality
- Quality Vs. Grade
- Quality Assurance (QA) Vs. Quality Control (QC)
- Cost of Quality (COQ) = Cost of Prevention + Cost of appraisal + Cost of failure (CoPQ).
- Static testing:
 - Software is not executed
 - Coding standards, Static code analyzer, Code reviews...
- Dynamic testing:
 - Software executed and tested
 - Functional (black box) testing, Structural (white box) testing

Quality Standards

- Agreed approach to ensure 'conforming to requirements'
 - The QA & QC Strategies
- Tools include:
 - Benchmarking
 - Cause and effect (Fish bone, Ishikawa) diagram
 - Root Cause Analysis (RCA)
 - Control charts
 - Pareto charts
 - Scatter diagram
 - Run chart
 - Flow Chart
 - Force field analysis: Looks at forces that are either driving movement toward a goal (helping forces) or blocking movement toward a goal (hindering forces)
 - Quality Checklists: is an instruction sheet for an inspector to verify that a set of required steps has been performed

Escaped defects

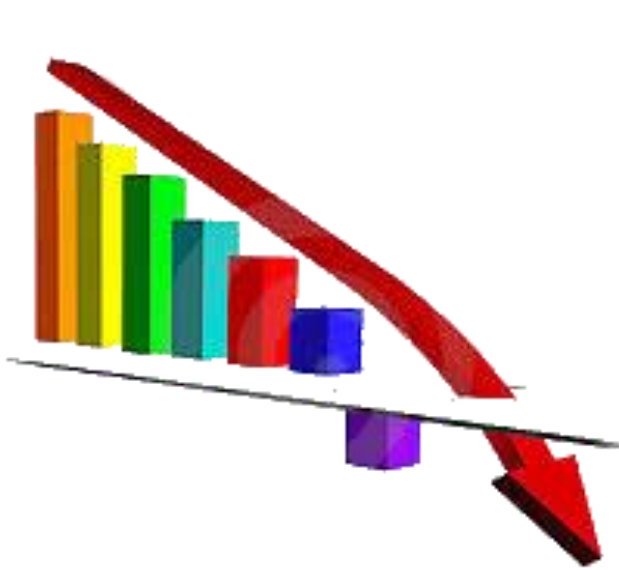
- Number of defects escaped to customer
- Typically measured against releases
 - But other time periods can also be tried
- This Feedback metric could be used to improve the quality over time

$$\text{Escaped defect rate} = \frac{\text{Number of escaped defects}}{\text{Total story points}}$$



Trend Analysis

- Leading Indicator
- Trend direction provides more info than absolute numbers
- Moving Average to tackle past history baggage



Brainstorming techniques

- Used for innovation, problem solving...
 - Requires Cross team, Open environment & good facilitator
1. Warm up / Ice-break
 2. Goal setting
 3. Share ideas: 'quiet writing', 'round robin approach', 'free for all' format
 - Postpone criticism
 4. Sort: cluster similar ideas, remove duplicates
 5. Rank
 - MoSCoW
 - Dot voting or Multi voting (members given ~20% points to distribute to items)
 6. Act / Solve

Failure modes and alternatives

- Relate to HUMAN side of Performance and Process
- Alistair Cockburn has listed 5 failure modes:
 1. Making Mistakes (human tendency !!)
 2. Preferring to fail conservatively (comfort zone syndrome)
 3. Inventing rather than researching (Re-inventing the wheel!)
 4. Being creatures of habit (Resist change)
 5. Being inconsistent (in adhering to process)
- Fortunately human have some counter behaviors like
 - Being good at looking around, Being able to learn
 - Being malleable, Taking pride in workbased on which Cockburn suggests 10 strategies for overcoming 5 failure modes

Failure modes and alternatives

1. Countering with discipline (standard practice) built in with some tolerance
2. Start with something concrete and tangible
3. Copying and altering (template, blank page syndrome)
4. Watching and listening (Expert in earshot)
5. Supporting concentration and communication (e.g. Caves and common, Quiet work period)
6. Personality matched work assignments
7. Talent (retain best talent)
8. Rewards that preserve joy
9. Complement rewards with other factors like good work environment
10. Feedback (Like continuous integration, pair programming, iteration reviews)

Knowledge Sharing

- Critical to Knowledge intensive IT projects
- Enablers:
 - Release / iteration planning
 - Daily scrum meetings
 - Retrospectives
 - Iteration Demo
 - Co-location (Osmotic Communication)
 - Cross functional team
 - On-site customer
 - Pair programming & Pair rotation
 - Common code ownership
 - Information radiator

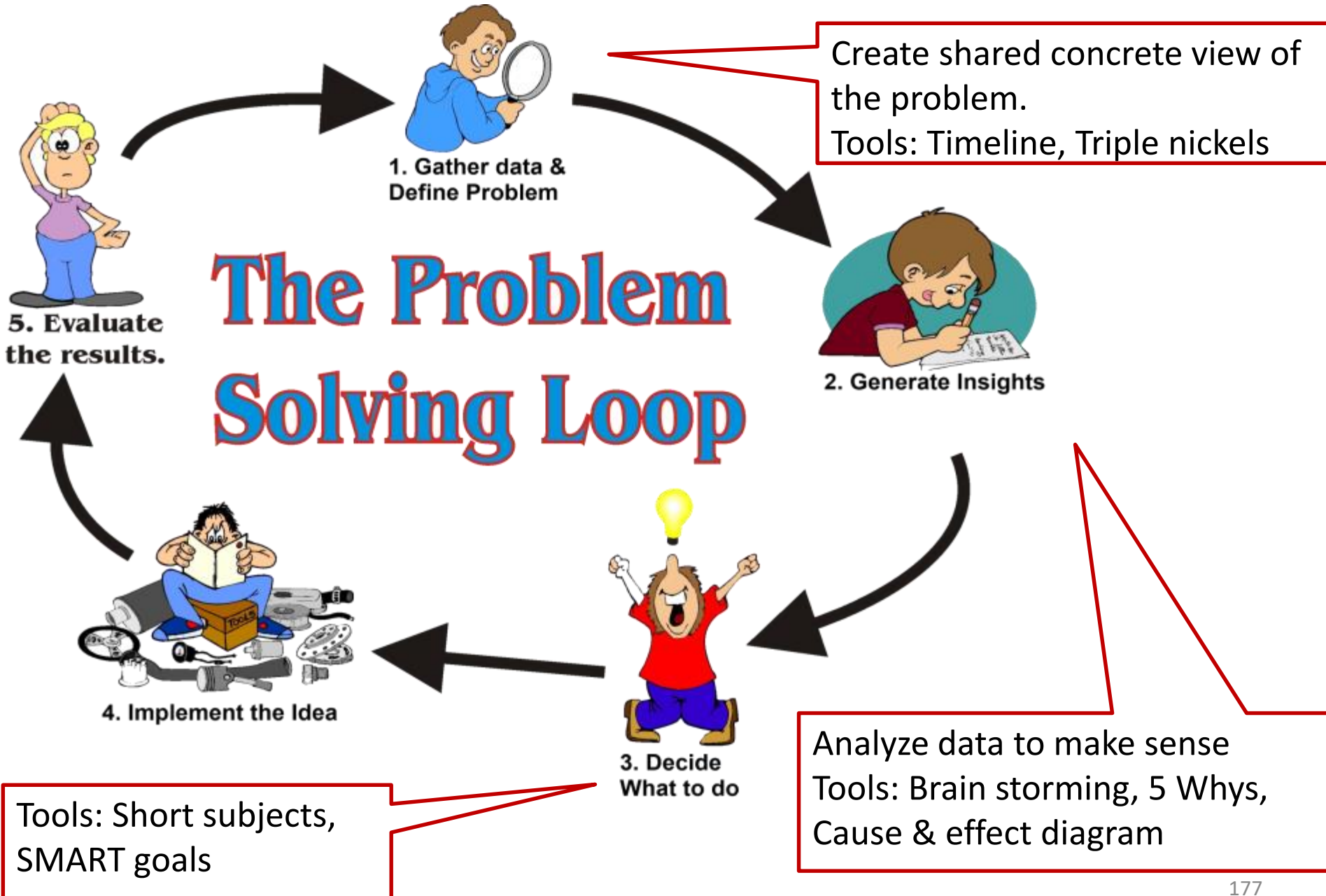
Knowledge Sharing: HR Practices

- Rewarding Individuals' competency & not for what they share
 - discourages knowledge sharing
 - Rather, reward the discovery, innovation & knowledge sharing
- ‘Measuring up’ refers to measuring at one level above the normal span of control
 - To encourage collaboration, team work, knowledge sharing, global (not local) optimization
 - Ex: Team level over Individual level; Velocity

Self assessment

- Comprehensive review of oneself to understand strengths & weaknesses and opportunities to improve
- The purpose is to stimulate learning & change, as well as enthusiasm for development
- SWOT analysis

Problem detection & resolution



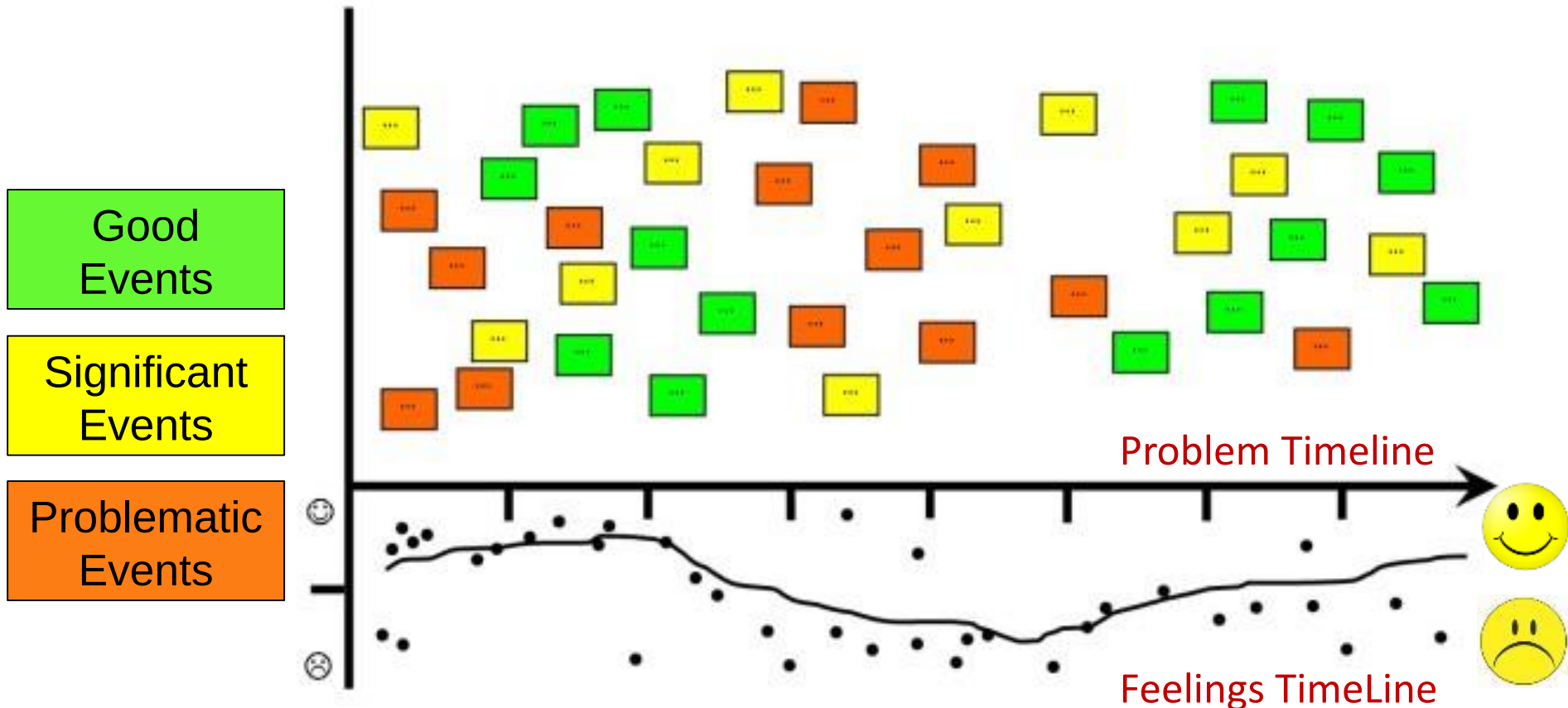
Problem detection & resolution

- Involve team:
 - Leverage on team knowledge
 - Get implicit team buy-in
 - Practical solutions
 - Team synergy



Gather data: Timeline

- Spatial representation of cause & effect; Used by doctors
- Individual identify, Paste it on timeline, Team discusses left → right
- Team also record below their feelings (good, sad...) about the events below the timeline in separate 'Feelings' timeline



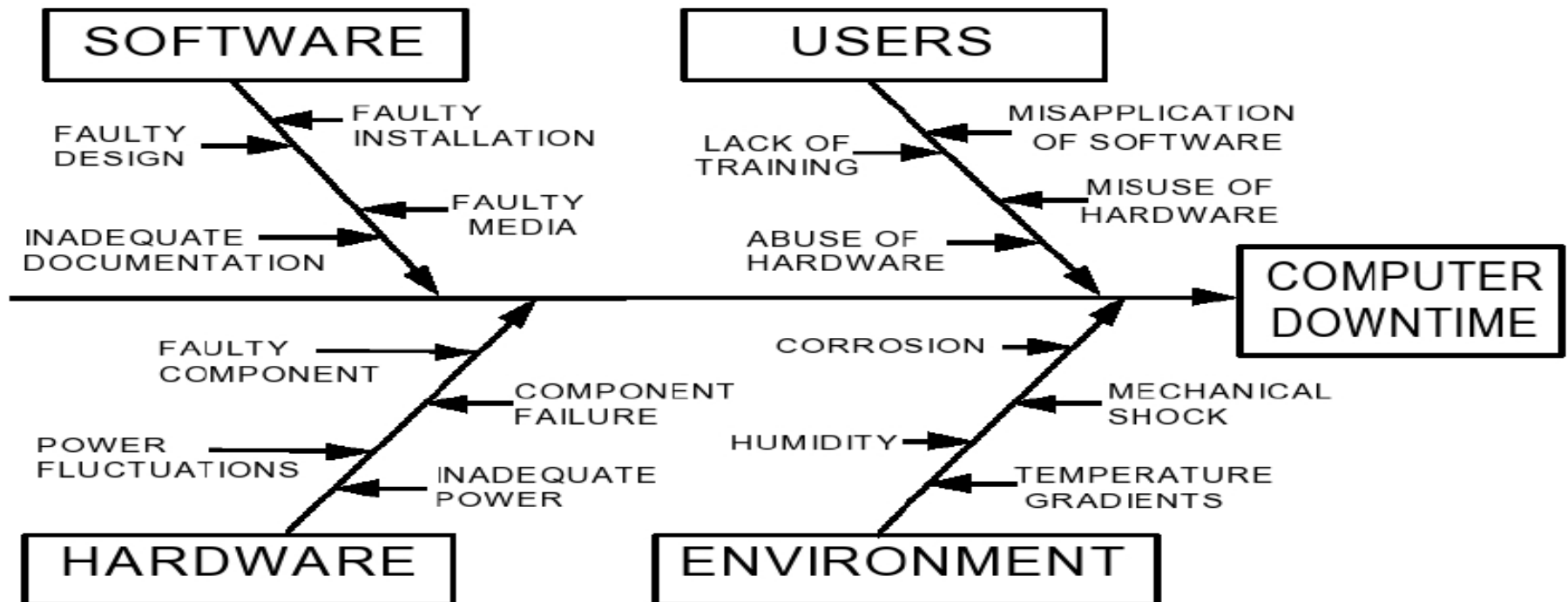
Gather data: Triple Nickels (5-5-5)



- Groups of 5 each
- Individual generate 5 ideas in 5 minutes
- Pass on ideas to others in round robin for additions
- Personal reflection and expanding others' ideas

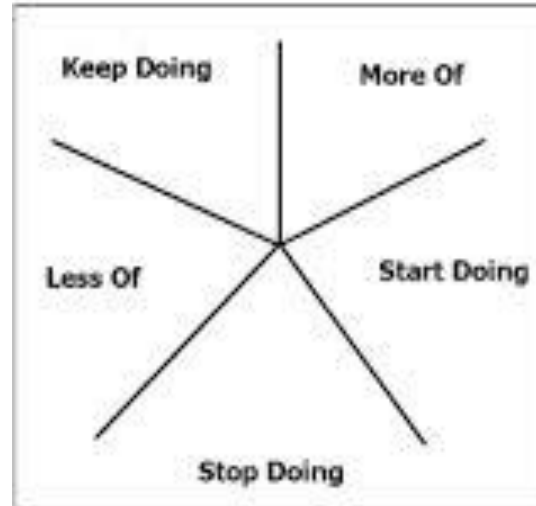
Generate Insights: Cause & Effect diagrams

- Also known as Ishikawa diagrams, fishbone diagrams
- To understand the causes of a specific event
- Keep problem statement at the head
- The causes are found by keep asking WHY until the ACTIONABLE root cause has been identified or until the reasonable possibilities on each fishbone have been exhausted



Decide What to do: Tools

- Short Subjects:
 - Team is presented with white board with CATEGORIES written on them to add
 - WHAT WENT WELL, DO DIFFERENTLY NEXT TIME
 - KEEP, DROP, ADD
 - START DOING, STOP DOING, DO MORE OF, DO LESS OF



- SMART goals:
 - Specific, Measureable, Attainable, Relevant, Timely
 - Essential qualities for an effective solution

15. AGILE ADAPTATION

Process tailoring

- Amending established methodology or standards to better fit the project environment
- Some methodologies are easy to tailor (Ex: Kanban) whereas others are not (Ex: Scrum)
- 2 schools of thought
 - Favors tailoring as every project is unique
 - Arbitrary tailoring will take the essence out
- Taking merits from both schools, we shall follow Shu-Ha-Ri:
 1. Get accustomed to out of the box AGILE, Understand it
 2. Examine the motivation for tailoring: Is it for convenience masking a fundamental problem OR value add OR addressing a gap?
 3. Involve appropriate stakeholders, discuss, decide, implement
- Retrospective reviews are good platform for driving process changes

Process analysis

- Related to process tailoring & the principles of systems thinking
- Reviewing and diagnosing issues with agile methods (or proprietary methods) to take a decision on process tailoring

Methodology anti patterns:

- One size fits all
- Intolerant / Not flexible
- Heavy
- Embellished
- Untried
- Used once (not tested on different weather)

Success signs:

- Deliverable(s) got shipped
- Leadership team intact (not fired)
- Team would work the same way again

Applying new agile practices

When you face a problem for which you find need for new agile practices, ensure below things

1. Does any existing agile (tested) practice helps? Do research first before invent
2. Go for tested practices in the successful projects after ensuring the project circumstances are similar to one in the tested project(s)
3. Adopting new practice is akin to taking new drug. Seek
 - a. Is there a natural solution? e.g. dropping of problematic practices
 - b. Before adopting less tested practices, investigate yourself... go through case studies, contact relevant project team...
 - c. Try small dose like in few iterations before broadcast it
 - d. Review the side effects too

Measuring Agile compliance / maturity

Sidky agile measurement Index (SAMI)

- Value-based agility measurement and 4 stage Agile migration framework

Agile Levels

1. **Collaborative:** Enhancing Communication & Collaboration
2. **Evolutionary:** Delivering SW early and continuously
3. **Effective:** Developing Quality, Working SW in an efficient an effective manner
4. **Adaptive:** Responding to change through multiple levels of feedback
5. **Encompassing:** Establishing a vibrant environment to sustain agility

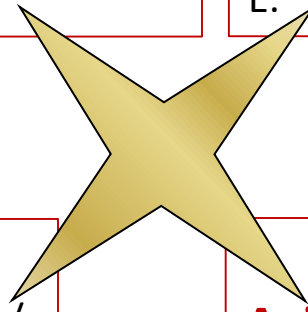
Agile Principles

- A. Embrace Change to deliver Customer value
- B. Plan & Deliver SW Frequently
- C. Human Centric
- D. Technical Excellence
- E. Customer Collaboration

Indicators

- Access Org / Project characteristic to access org / project 'agile' readiness
- People, culture, and environment

Agile Practices and Concepts



PUTTING IT ALL TOGETHER

Agile hugely helps

- When you have high requirements volatility
- When the technology is new / complex
- Delivering HIGH values first
- Right shaping by frequent iteration demos
- Quicker TTM & Higher ROI
- Fast failure on infeasible projects

Value to all

Customer

- Incremental delivery
- Client focus, Client drive
- Change friendly



End user

- Incremental release
- Valuable features faster
- Potential influence during development



Organization

- Less escalation as customer drives the project
- T&M: Predictive Revenue, No buffer overhead
- Simple process
- Possible new business



Team

- Empower, Trust, Respect
- Self organization
- Experience for future leader
- Cross Technology / domain Training
- System (big picture) thinking



Sponsors

- Predictive demand as end users are part of requirements
- T&M: Predictive cost
- High ROI, Early Revenue



GETTING STARTED WITH AGILE JOURNEY

Agile Shift

ABO Continuum

- 1998 Arthur Andersen

OWNERSHIP

Team try,
believe,
adopts, owns
the change

BUY-IN

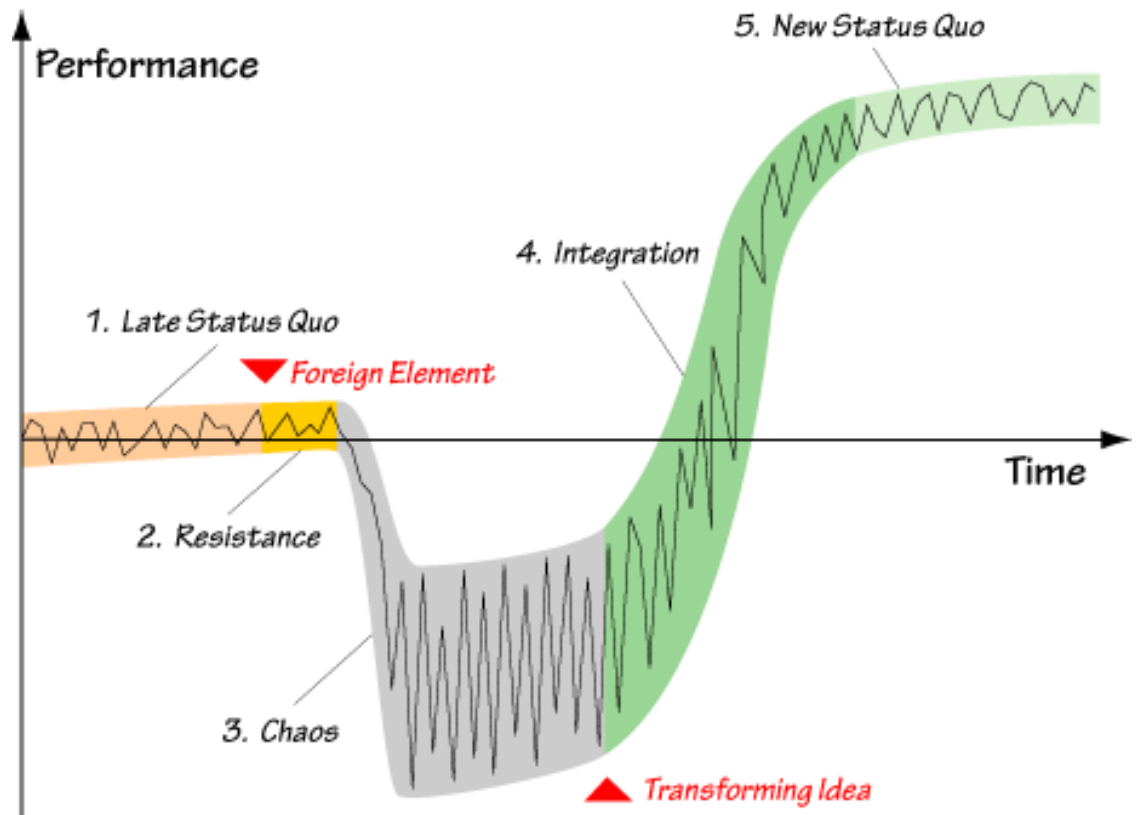
Roll-out the
change

AWARENESS

Prior informal
intimation about
impending change

Virginia Satir Change Process Model

- Psychological model
- Popular in 1990-2000



Agile Shift

- Organizations trying to implement agile but without making the AGILE shift
- Scrum-butt: Agile practices don't fail; rather the variations on agile adaptation fail
- Be(a)ware of Agile anti-pattern: e.g.
 - Old habits hardly die
 - “My boss is on my team”
 - “Our daily scrum lasts 45 minutes”
 - “We’re doing a documentation sprint before we start coding”

Agile Shift

- Pilot Project:
 - Preferably internal customer, project that hits all phases, medium priority project
- Follow Shu-Ha-Ri:

Shu Ha Ri

3. Examine the motivation for tailoring:
Is it for convenience masking a fundamental problem OR value add OR addressing a gap?
Involve appropriate stakeholders, discuss, decide, implement

Be the rule

MASTER

Own Practice,
Create
approaches

離
Ri

Break the rule

APPRENTICE

Understand
underlying
principles

破
Ha

Follow the rule

STUDENT

Learn from
a Master

守
Shu

2. Understand agile

1. Get accustomed to
out of the box AGILE

Q&A TIME

THANK YOU