

Managing Business Analytics Projects

Development, Transition, Roll-out, and Maintenance

Richard Tan

Digital Products & Platforms

Total Slides = 80+24

© 2015-2019 NUS. The contents contained in this document may not be reproduced in any form or by any means, without the written permission of NUS, other than for the purpose for which it has been supplied.

Module Objectives

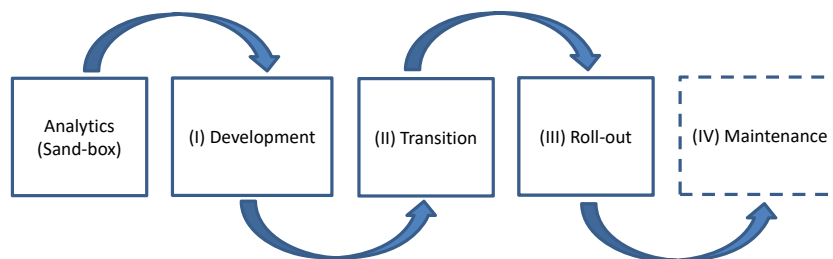
- To further develop the requirements and manage the development-transition-rollout and maintenance phases of a business analytics projects



Agenda

- The Uniqueness of Analytics Projects
- Development
- Transition
- Rollout
- Maintenance

Typical Analytical Project Phases

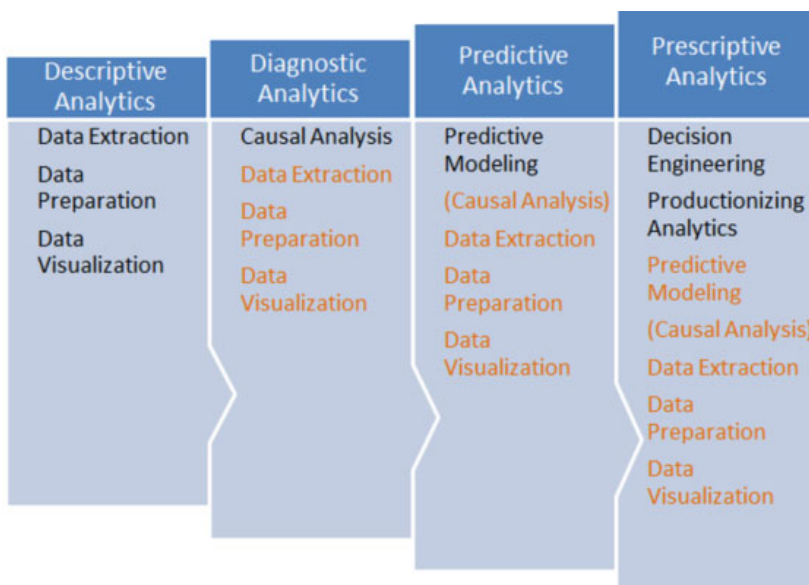


- A complex variety & hybrids of building blocks
- Must understand your requirements...

Business Analytics Complexities

- Data Analytics variety
- Business Analytics frameworks
- Implementation Frameworks
- Data & technical complexities
 - Structured or unstructured; Static or Streaming; Big Data
- Rollout complexities – local or regional / geographical
- Insource or Outsource
- Technical infrastructure – on-premises or cloud
- DevOps
- Human Resources

Variety of Business Analytics



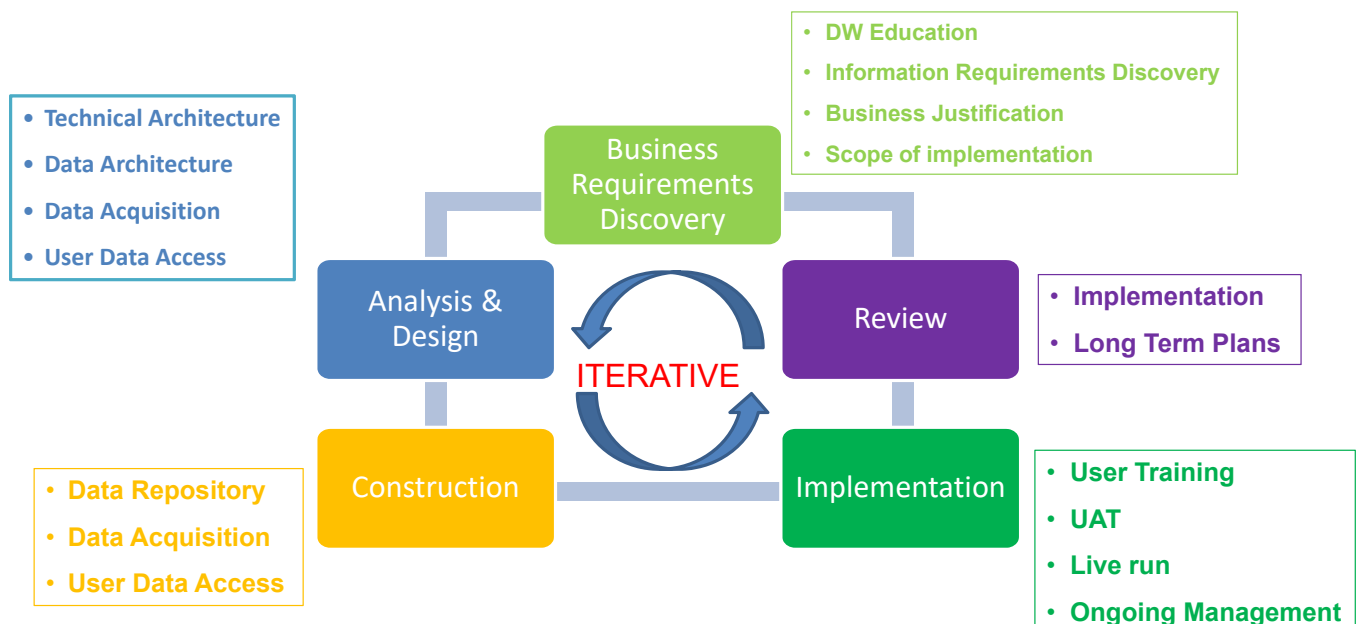
- Data Visualization
- Slice & Dice Capabilities
- Standard (pre-canned) Reports
- Ad-hoc Reports
- Data mining
- Prescriptive – Hypothesis???
- Machine / Cognitive Learning
- ...

Business Analytics Frameworks (1/2)

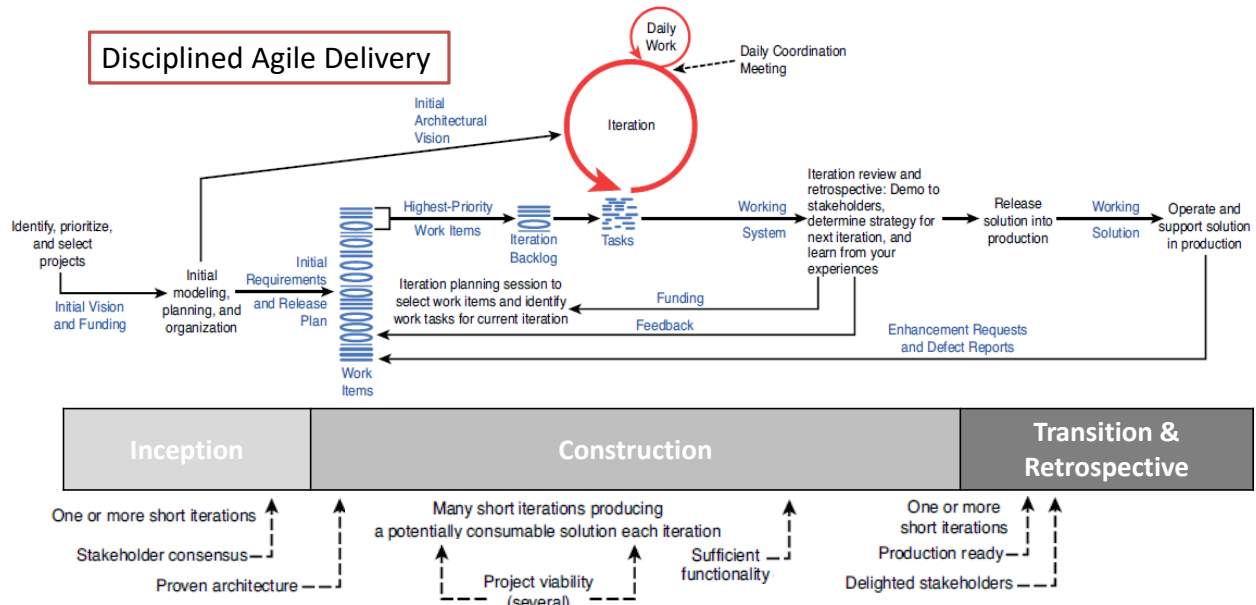


CRISP DM Framework

Business Analytics Frameworks (2/2)



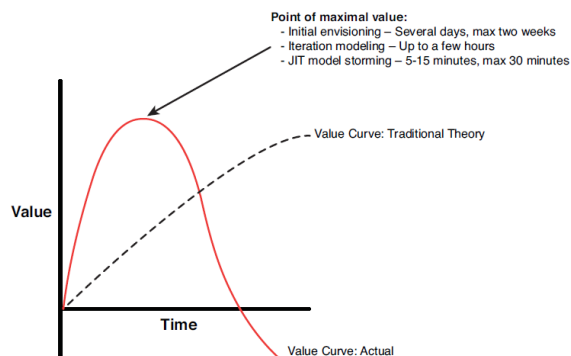
Implementation Frameworks



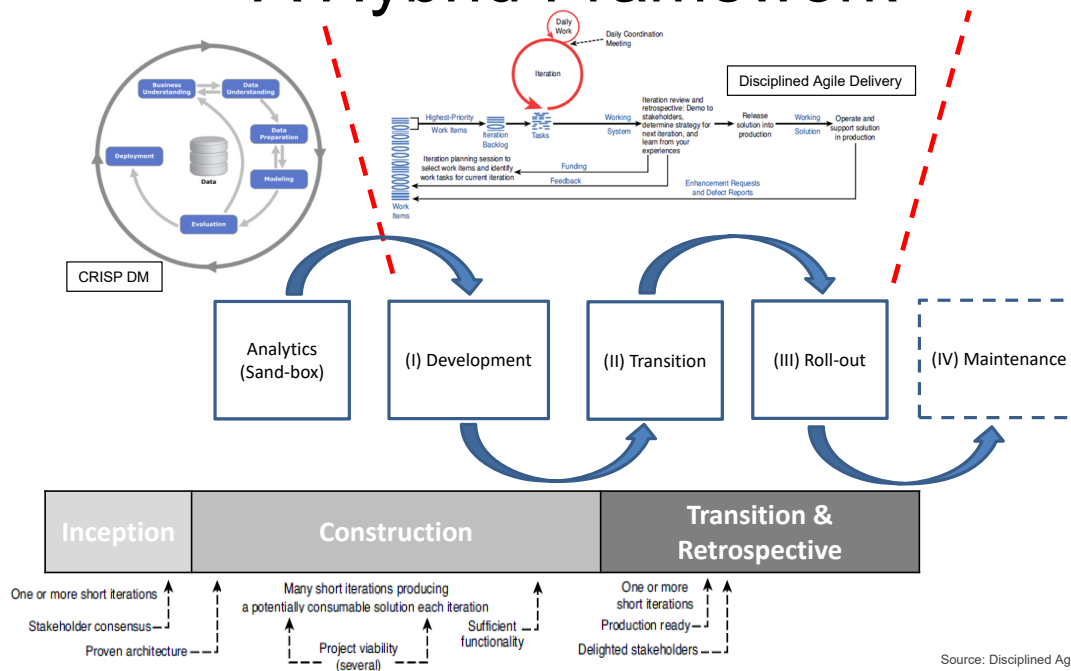
Source: Disciplined Agile Delivery; Scott W. Amber; Mark Lines

Project Delivery Mantra...

- Deliver Value – early & often
- Value is what your enterprise or stakeholder is willing to pay for a product or service to:
 - Produce a benefit
 - Improve a service

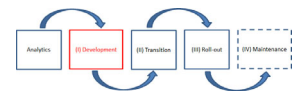


A Hybrid Framework



Source: Disciplined Agile Delivery, Scott W. Ambler, Mark Lines

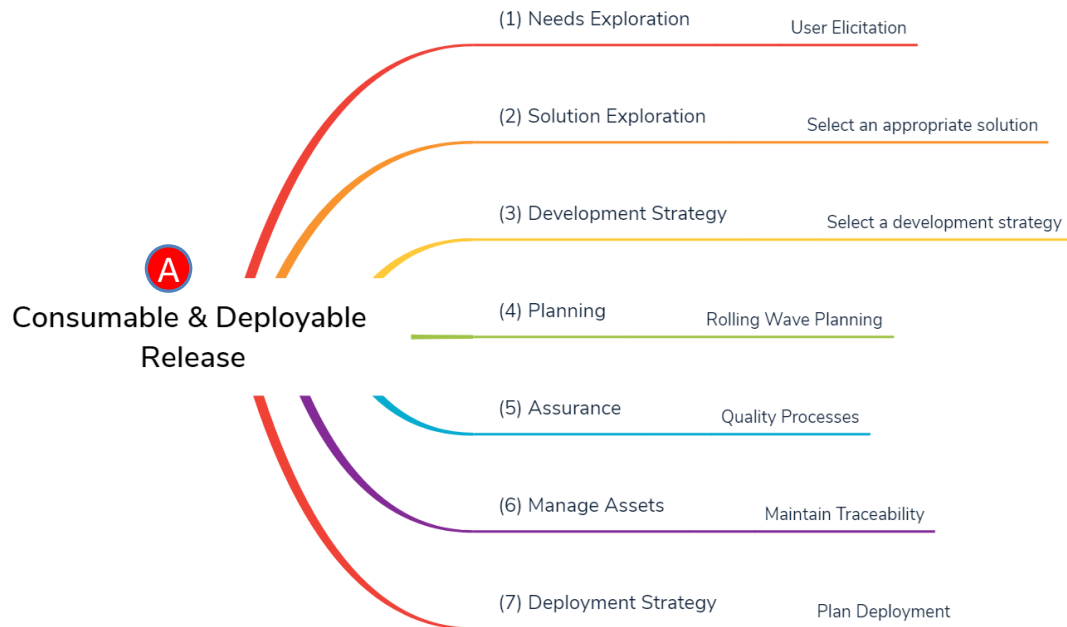
(I) Development - Objectives



- A** Produce a Consumable & Deployment Solution
- B** Address Changing Stakeholder Needs
- C** Prove Your Architecture Early
- D** Tracking & Control

➤ **Produce a solution that will meet or exceed your stakeholder's needs & expectations**

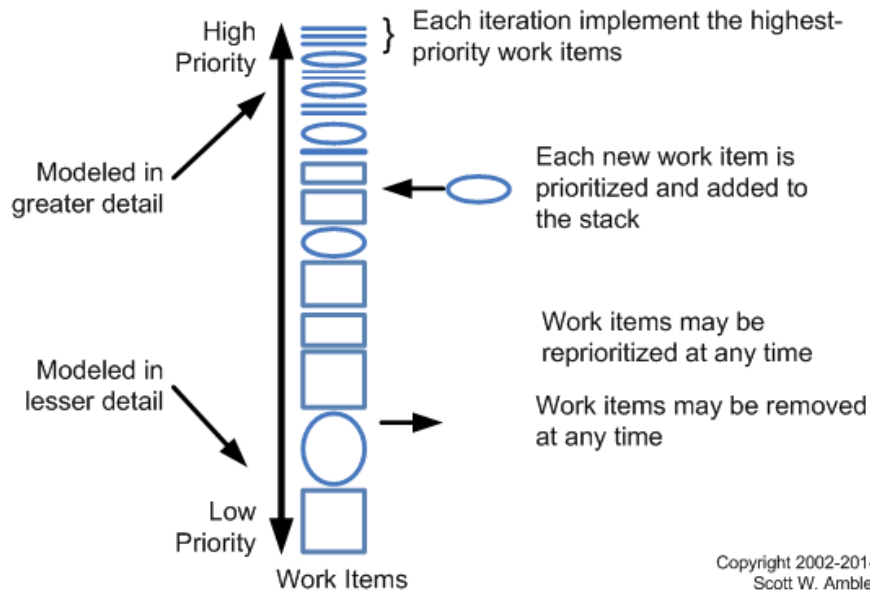
Consumable & Deployable Release



(1) Needs Exploration

- Break down the initial scope according to stakeholders' priorities and concerns
- Sub-divide the major project deliverables into smaller more manageable components to:
 - Ensure delivery with the specified features and functions
 - Improve the accuracy of scope management, cost, time and resource estimates
 - Have a baseline for performance measurement and control
 - Facilitate clear responsibility of assignments
- The outcome will be the "Work Item Stack / List"

Work Item Stack



User Story Considerations

- Adopt the iterative and agile approach
- Identify a handful of / sufficient goals
- Elicit sufficiently to ensure:
 - Stakeholders' agreement to what will be delivered
 - Ability to understand **value & scope**
- Further details should be explored in a Just-in-Time (JIT) manner during construction
- Practice high-level design constraints
- Allow the details to emerge over time
- Elicitation effort should take hours or days – not weeks or months

User Story – The WHAT

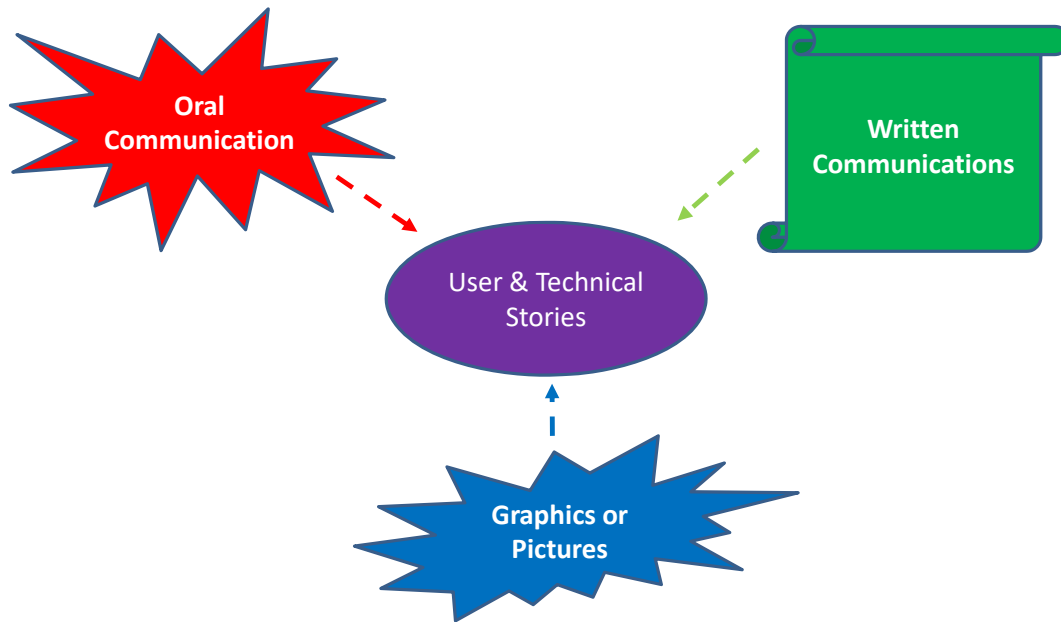
- Similar to a use case – describes a feature your user needs the product / service to do **NOT** how to implement it
- It should be a unit of work that your team commits to complete in a **release or an iteration**
 - Sub-tasks (if any) should be decided by the team
- Exclude technical aspects
- Use **SMART** criteria to develop tasks
- Allows your team to focus on the valuable outcome to your stakeholder

Adapted source: Bill Wake; <http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>

Technical Stories

- Targets the **non-functional features or business-related capabilities** to support for the business functional behaviour
 - Ensures alignment with your enterprise architecture requirements
 - Typical activities include technical analysis, design, solution architecting work, prototyping...
- Examples: security, performance, scalability, availability...

Critical Criterion...



Sample Template

- 1) Who is the user we are building for?
As a <type of user>
- 2) What is the goal or objective of what we are building?
I want <some goal or objective >
- 3) What value does it bring to the user?
So that <benefit, value>

User Story - Example

- As a vice president of marketing, I want to select a holiday season to be used when reviewing the performance of past advertising campaigns so that I can identify profitable ones.
- Detail could be added to that user story example by adding the following conditions of satisfaction:
- Make sure it works with major retail holidays: Christmas, Easter, President's Day, Mother's Day, Father's Day, Labor Day, New Year's Day.
- Support holidays that span two calendar years (none span three).
- Holiday seasons can be set from one holiday to the next (such as Thanksgiving to Christmas).
- Holiday seasons can be set to be a number of days prior to the holiday.

Source: <https://www.mountaingoatsoftware.com/agile/user-stories>

Technical Story - Example

“We need to extend the kiosk authentication code in our security services layer to include a new authentication mechanism for web-based (browser) applications. It needs to include 2-layer authentication: passwords and user-centric questions.”

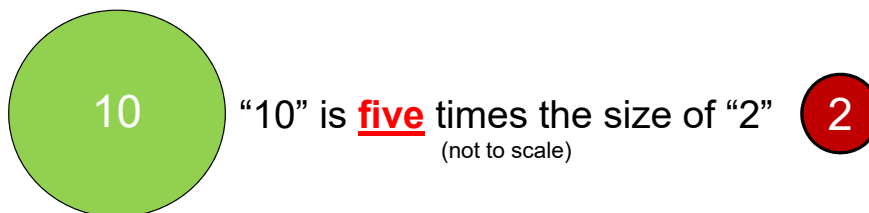
A typical User Story format:

As a user requesting authentication, I need to be able to login via the web app, so that I can manage my account details via the web

Source: <http://rgalen.com/agile-training-news/2013/11/10/technical-user-stories-what-when-and-how>

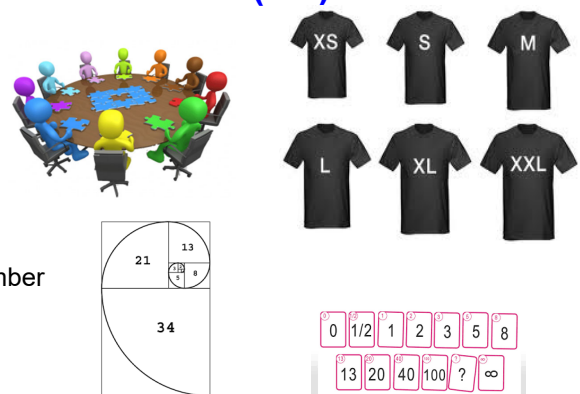
Story Estimation Methods (1/2)

- Use of **Story Points** – it is “unit-less”
- It is objective - minimizes biases
- It is democratic - facilitates consensus building
- Allows comparison between multiple user stories
- Estimation is done by a team member who is implementing it OR someone with superior knowledge about the work → **experiential estimation**



Story Estimation Methods (2/2)

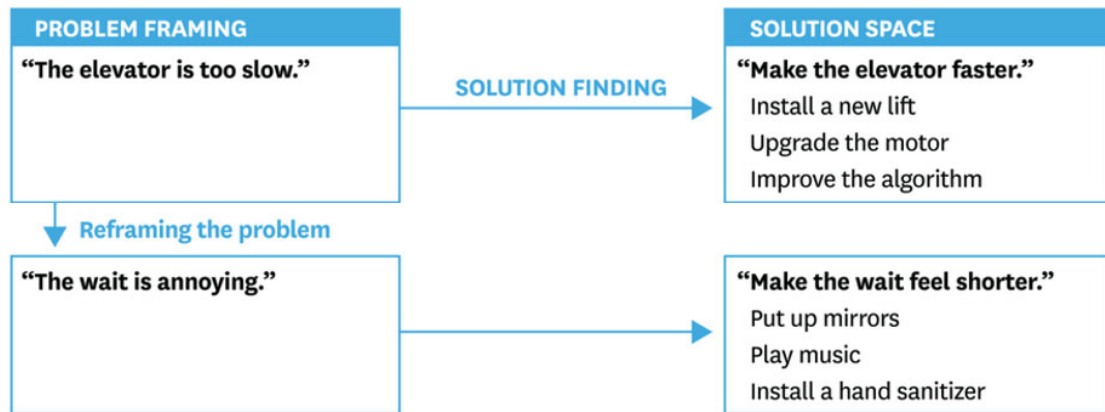
- **Relative Sizing aka “Tee-Shirt” sizing**
 - Start with Smallest
 - with the **smallest** user story; assign 1 point to it
 - Estimate every other user story in relation to that
 - Start in the Middle
 - with a user story of **middle** size, and assign the middle number on your scale to it
 - Estimate every other user story in relation to that
- **Planning Poker or SCRUM Poker**
 - “Fibonacci Numbers”
- **Use “Triangulation” to compare the sizes of different stories**



	Story points
User Story 1	2
User Story 2	6
User Story 3	1
User Story 4	?

(2) Solution Exploration

- Understand the problem you are trying to solve...
- Frame the Solution



Source: <https://hbr.org/2017/01/are-you-solving-the-right-problems>

Explore Available Solutions

- Best of Breed or All-in-One
- Buy or Build - COTS or Bespoke
- Insource or Outsource
- On-premises or in the Cloud

➤ Always look both ways – in and out

(3) Development Strategy

Development strategy considerations:

- Decide the development strategy
- COTs or Bespoke
- In-house or outsource or hybrid
- Various types of agile development strategies:
 - ATDD – Acceptance Test Driven Development
 - BDD – Behavior Driven Development
 - TDD – Test Driven Development
 - FDD – Feature Driven Development
- Organize a delivery cadence

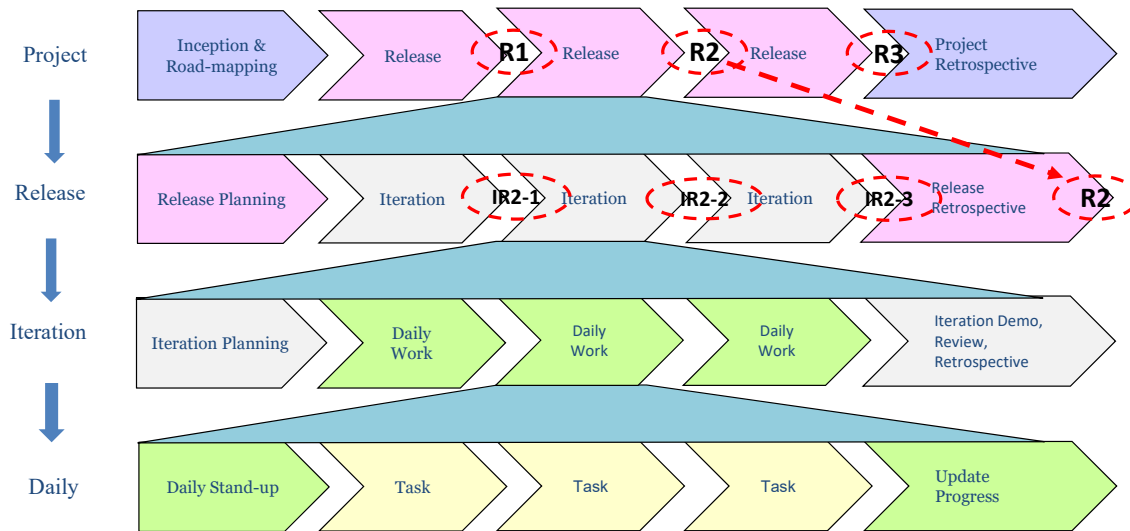
(4) Planning

- Organize your project into a set of iterations
- Planning strategy is required
 - Must have a potentially consumable increment of the solution being built & demonstrated
- Must have cadence or a regular iteration of release
- Release a working build when it is appropriate
- Estimate the cost & value
- Allows risk identification and its management



Source: Disciplined Agile Delivery Scott W. Ambler; Mark Lines

Rolling Wave Planning



(5) Assurance

- Quality Processes & Standards
- Improve the quality
 - System
 - Documentation
 - Leverage enterprise professionals
 - Enterprise architects, Security
 - Application Development teams, PMO, DevOps, ...
- Risk Management

A Typical Scenario...

User Story: “The Employee’s Salary information (read access) is accessible by the said Employee”

Project Manager: Is this function done?

Developer: Yes!

Project Manager: So we can ship it?

Developer: Err...no. It needs to be tested. I need to write some documentation...but the code really works. I tested it... (pause) ...on my machine.

➤ **Are multiple definitions of “done” suitable for you / team?**

Some Definitions of Done (1/2)

- What is your definition of Done?
- It is a shared understanding of what it means when work is considered as done.
- Posted in a visible area (Big Visible Chart) in the team’s shared workplace
- Brings transparency to the team and help prevents “undone” work
- Ensures stakeholders’ expectations are met!

Some Definitions of Done (2/2)

User Stories (US)

- All codes have been checked in
- US unit tests passed
- US integration tests passed
- US acceptance tests passed
- The US are approved by Business Owner

Iterations

- All User Stories in sprint passed acceptance criteria of stakeholder
- The source code has been reviewed.
- All tests were successfully completed
- Regression tests passed
- Performance tests passed

Transition

- Stress tests completed
- Security audit finalized
- Relevant Documents updated
- The build is in a release-ready state and available for download

A Shirt (as a Product) Example...

Are we building the "Product" right?

- Does it have 2 sleeves?
- Are the sleeves long?
- Are the collars button down?
- Is its size "Large"?
- Is it "Blue" in color?
- Are there any missing buttons?



Are we building the right "Product"?

- Does the collar fits me?
- Do the sleeves fit the length of my arms?
- Is it comfortable to wear?
- Can I afford it?
- Is the material of good quality?
- Is the shirt blue in color?
- Will my spouse like it?

The IT Product Example (1/2)



- 1) Are you building the product right?
- 2) How do you verify?

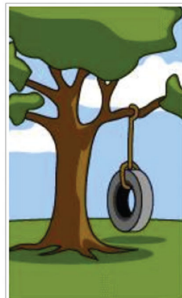
Your User Story requires
this Product to be built

The IT product Example (2/2)

Don't build this ...



if all you need is this.



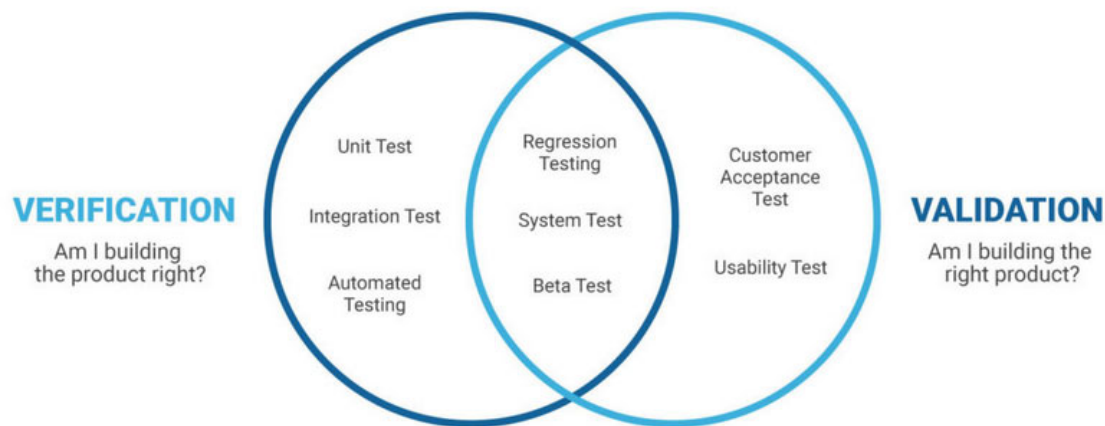
- 1) Are you building the Right Product?
- 2) How do you validate?

Avoid:
IKIWISI
Syndrome

Leverage the concepts of MVP:

- Implement functionality when you need them
- Do the simplest thing that could possibly work

The Importance of Testing...



Source: <https://www.plutora.com/blog/verification-vs-validation>

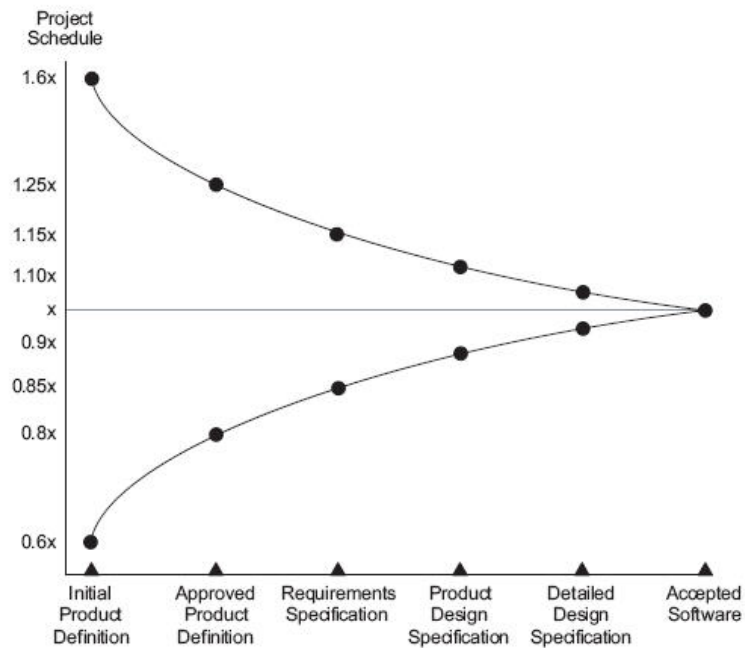
Refer to Appendix 1: Verify versus Validate

Managing Business Analytics Project Risk

- Risks are identified and handled as part of daily standups and also during reviews & retrospectives.
- All planning meetings must consider risks involved in the iteration carefully.
- Sometimes a separate risk plan is done and a risk burn down chart is created.
- Standard risks like time to market, budget risk, requirements error, scope creep, technology risks etc. can be tracked.

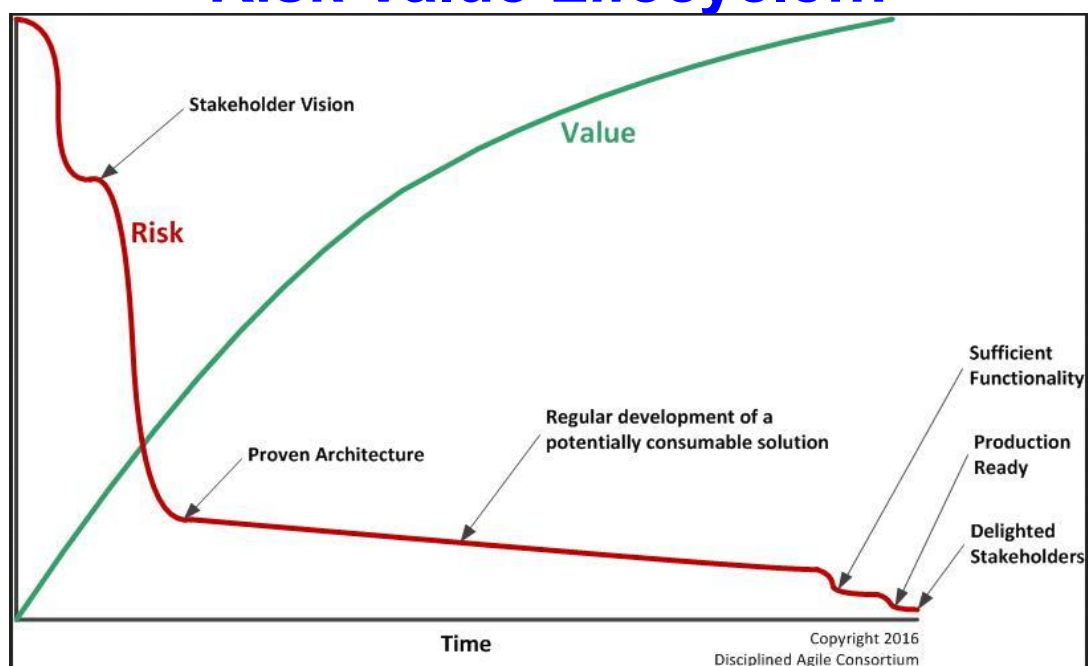


Uncertainties of Development...



Source: Cone of Uncertainty; Barry Boehm, 1981

Risk Value Lifecycle...



Copyright 2016
Disciplined Agile Consortium

Typical Risk Items...

- What are some typical project risks?
- Any peculiar business analytics risks?

Risk Analysis

- Step 1: Identify the risks & capture the risks in the Risk Backlog
- Step 2: Determine severity of risks
Risk Exposure = Probability x Impact
- Step 3: Develop the mitigation steps
- Step 4: Record, track & monitor the risk progress
- Step 5: Plot risk burn down chart

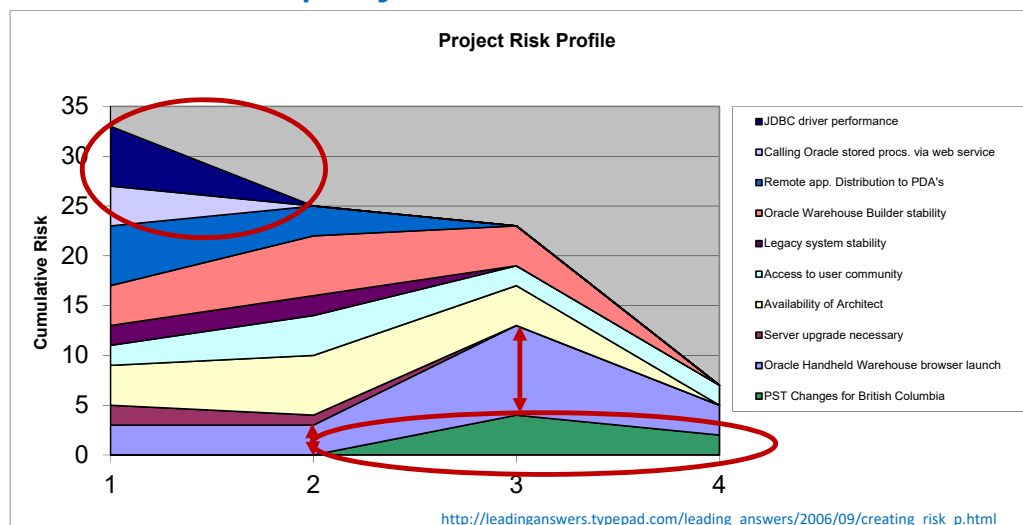
Tracking Risks

			Jan.			Feb.			Mar.			Apr.	
ID	Short Risk Name	Impact	Prob.	Sev.	Impact	Prob.	Sev.	Impact	Prob.	Sev.	Impact	Prob.	Sev.
1	JDBC driver performance	3	2	6	3	0	0	3	0	0	3	0	0
2	Calling Oracle stored procs. via web service	2	2	4	2	0	0	2	0	0	2	0	0
3	Remote app. Distribution to PDA's	3	2	6	3	1	3	3	0	0	3	0	0
4	Oracle Warehouse Builder stability	2	2	4	2	3	6	2	2	4	2	0	0
5	Legacy system stability	2	1	2	2	1	2	2	0	0	2	0	0
6	Access to user community	2	1	2	2	2	4	2	1	2	2	1	2
7	Availability of Architect	2	2	4	2	3	6	2	2	4	2	0	0
8	Server upgrade necessary	1	2	2	1	1	1	1	0	0	1	0	0
9	Oracle Handheld Warehouse browser launch	3	1	3	3	1	3	3	3	9	3	1	3
10	PST Changes for British Columbia	0	0	0	0	0	0	2	2	4	2	1	2
				33			25			23			7

http://leadinganswers.typepad.com/leading_answers/2006/09/creating_risk_p.html

Risk Burn Down Graph

Cumulative project risk severities over time

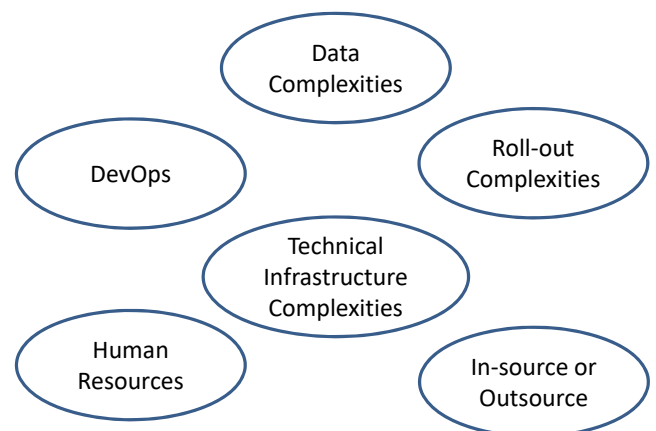


(6) Manage Assets

- How will we manage the assets that we produce?
- How will we document our work?
- How will we deploy our solution?
- Do we need to maintain traceability, and if so how will we do it?

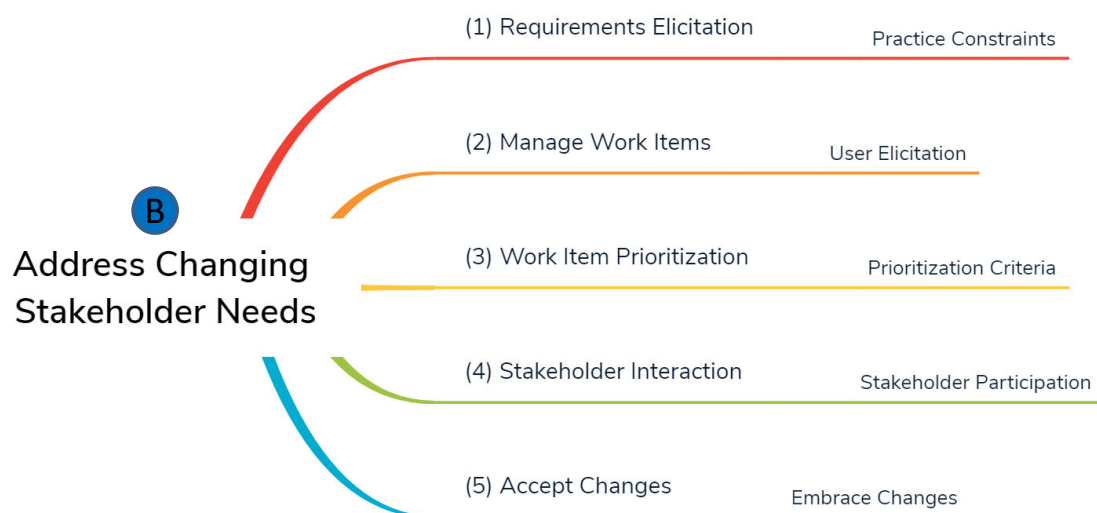
(7) Deployment Strategy

- Plan the cadence for deployment
- Big Bang approach or pilot deployment
- Work with DevOps
- Setup helpdesk
- Prepare fit-for-use documentation
- Conduct fit-for-use training



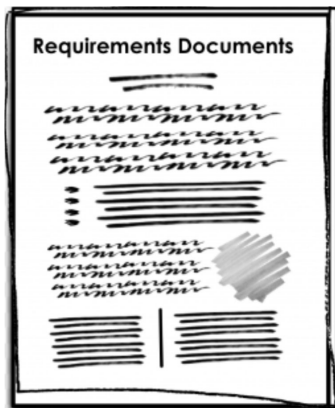
Workshop Part 2

Address Changing Stakeholder Needs



(1) Requirements Elicitation

Traditional



Agile



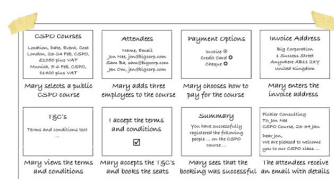
wireframes



personas



user stories



storyboards

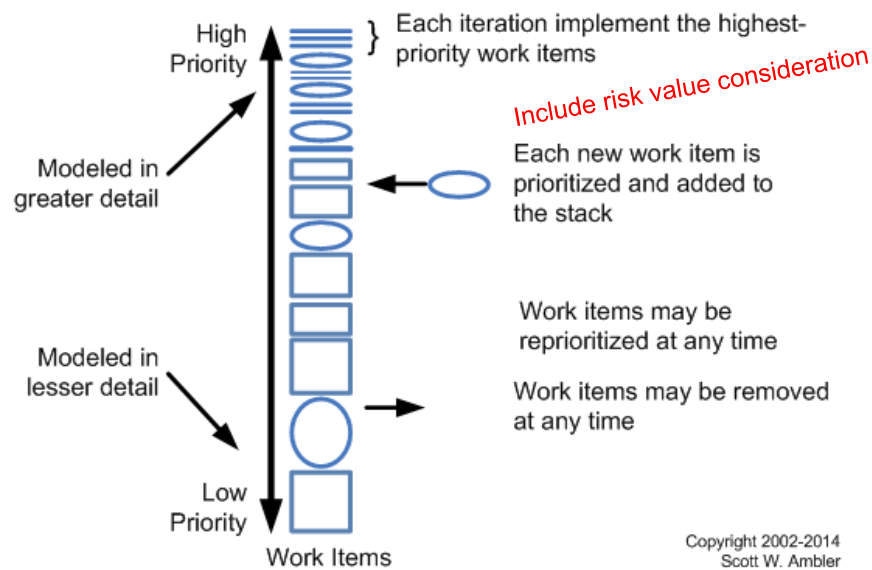
Some Elicitation Techniques...

Remember The BRUF???

- Standish Group - Chaos Report:
 - About 45% of functionality delivered is never used
 - About 19% is rarely used
- Wastage is inherently due:
 - Human behaviour - not good at defining things upfront
 - Change management processes tend to de-motivate users
 - Tendency to "**gold-plate**" - makeup more requirements than necessary



(2) Manage Work Item Stack



➤ Manage the changes – Not prevent it!

(3) Work Item Prioritization

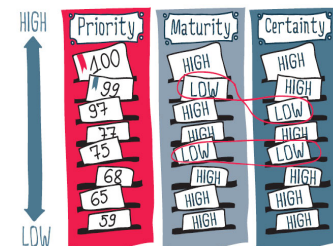


Prioritization Precedes Conflicts...

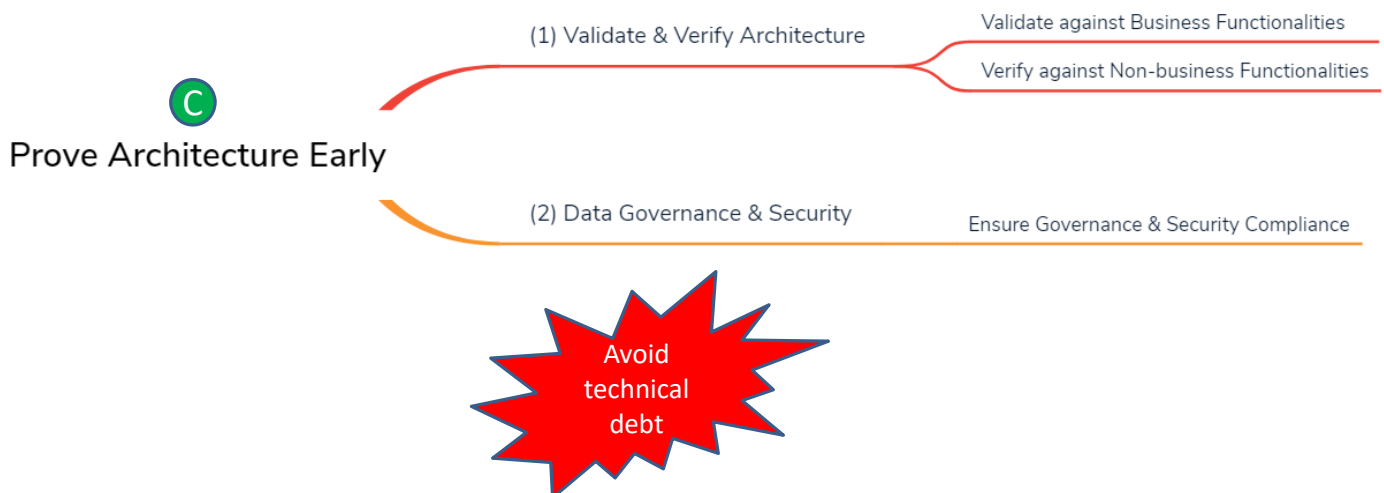
- “Everything is High Priority!”
- “How can you call this Low Priority?”

Possible Resolution Actions

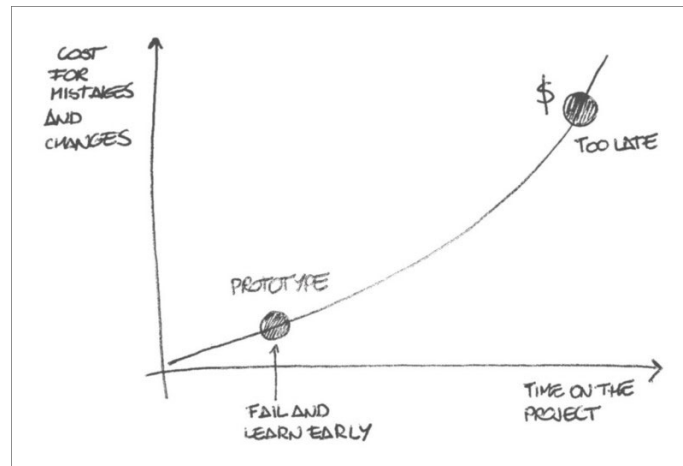
1. Use a combination of prioritization techniques
 - ✓ Business Value & Risk
 - ✓ Operational emergency dependency
 - ✓ Due Date
 - ✓ Weighted shortest job first (WSSJF)
2. Make it visual, tactile, collaborative
3. Impose forced limits / constraints



Prove Architecture Early



(1) Validate & Verify Architecture



“Fail Fast”; “Fail Early”; “Fail Often”

Architecture Considerations

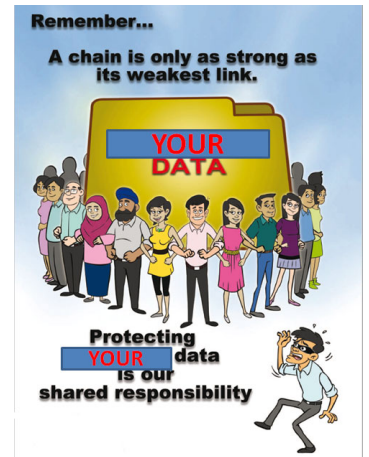
- Work with other enterprise professionals within your organization like EA; DevOps; Security; Governance etc.
- Leverage existing infrastructure as much as possible to allow their solution to evolve the infrastructure to reflect your organization's strategy
- Identify how much architectural modeling & documentation will be required
- Data preparation – includes acquisition, manipulation, storage
- Technical debt avoidance – could include NFR; security
- Review technical risks



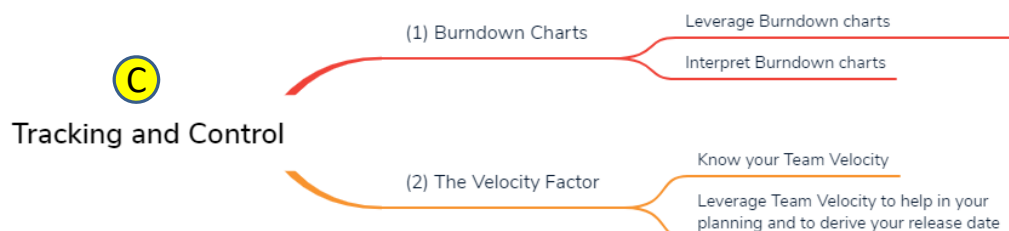
Source: John Donne's Devotions (1624)
"No man is an Island, entire of itself; every man is a piece of the Continent, a part of the main."

(2) Data Governance & Security

- 1) How important is data governance & security?
- 2) What is our weakest point?
- 3) What do we look out for...what are some of our considerations?
- 4) How do you ensure compliance?



Tracking and Control

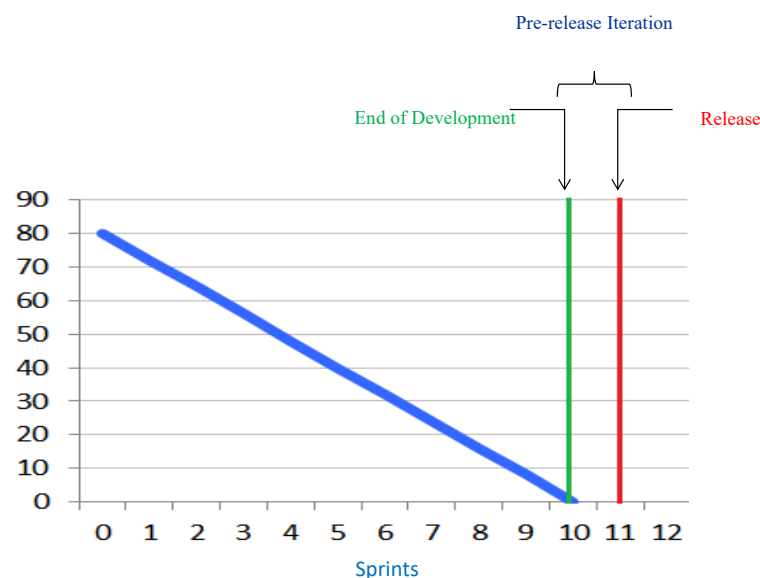


Tracking & Control

- Are we on track to deliver what we committed to for the iteration? (Iteration burn down)
- How much value have we delivered to business? (Release burn down; story points)
- How much value can the team deliver with remaining time? (Story points)
- As we plan for the next iteration, how much work can we commit to? (Team Velocity; story points per iteration)

Source : Metrics and Databases for Agile Software Development Projects by David Heimann

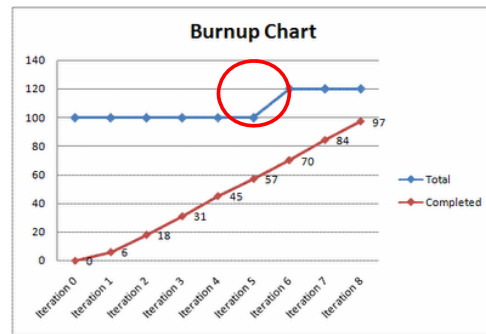
(1) Burn Down Chart



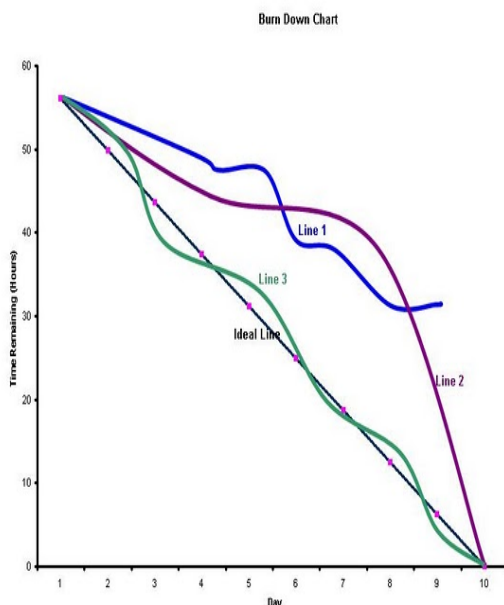
A graphical representation of “work left to do” versus time.

Burn Up Chart

- Tracks progress of work toward a goal line associated with a value on the vertical axis
- Curve is going up, not down
- Advantage:
 - Visible scope change



Interpreting Burn Down Charts



Line 1:

- The team planning is not good since their line never touches Zero!

Line 2:

- This line suggests that the team did reach its goal but it is fishy.
- Probably they abandoned a few user stories, overworked but didn't mention extra hours spent etc.

Line 3:

- This line suggests a burn down for a mature team who has been planning well, is self organized and have done enough stories to last the sprint.

Line 4:

- If actual line falls on the ideal line itself, then this is not a SW development project.
- There is something wrong – investigate further!

Work Item Stack

Requirements for a system, expressed as a prioritized list of work items stack

Priority	Description	Story Points (Estimates)
1	Configure machine	7
2	Track inventory	3
3	Purchase item	3
4	Validate purchase	2
5	Notification	5
6	Sales reports	7
7	Configurable Denominations	1
8	Change leveling	14
9	Advertising	10
10	Credit Card and Debit Payment	30
Total:		80

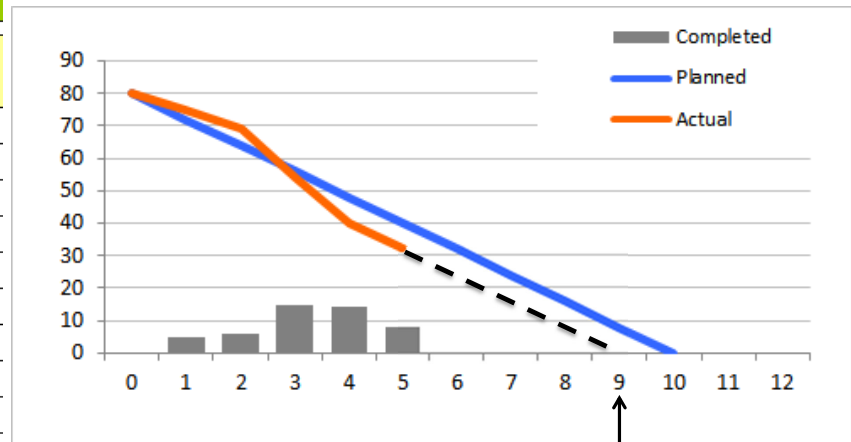
Tracking Release Items

S/No	Product Backlog Item	Estimates	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sprint 7	Sprint 8	Sprint 9	Sprint 10	Sprint 11	Sprint 12
1	Configure machine	7	2	0	0	0	0							
2	Track inventory	3	3	0	0	0	0							
3	Purchase item	3	3	2	0	0	0							
4	Validate purchase	2	2	2	0	0	0							
5	Notification	5	5	5	0	0	0							
6	Sales reports	7	7	7	1	0	0							
7	Configurable Denominations	1	1	1	1	0	0							
8	Change leveling	14	14	14	14	2	0							
9	Advertising	10	10	10	10	10	4							
10	Credit Card and Debit Payment	28	28	28	28	28	28							
Total:		80	75	69	54	40	32							

Updated Burn Down Chart

Burn down chart

Burned down			Balance		
Sprint	Planned	Actual	Planned	Actual	Completed
0			80	80	#N/A
1	8	5	72	75	5
2	8	6	64	69	6
3	8	15	56	54	15
4	8	14	48	40	14
5	8	8	40	32	8
6	8		32	#N/A	#N/A
7	8		24	#N/A	#N/A
8	8		16	#N/A	#N/A
9	8		8	#N/A	#N/A
10	8		0	#N/A	#N/A
11					
12					

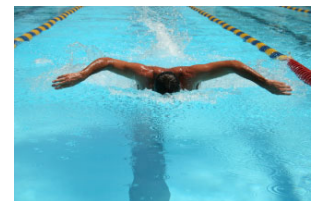


(2) The Velocity Factor

- It is used to plan the release date, with additional buffers for uncertainty and risk
- Measures how fast the team is delivering functionality in an iteration
- Velocity is calculated at the end of the iteration by totaling the points for all fully completed User Stories

Important note:

- 1) Always measure the rate of progress of the team.
- 2) NEVER measure for an individual member of the team!



The Velocity Factor - Example

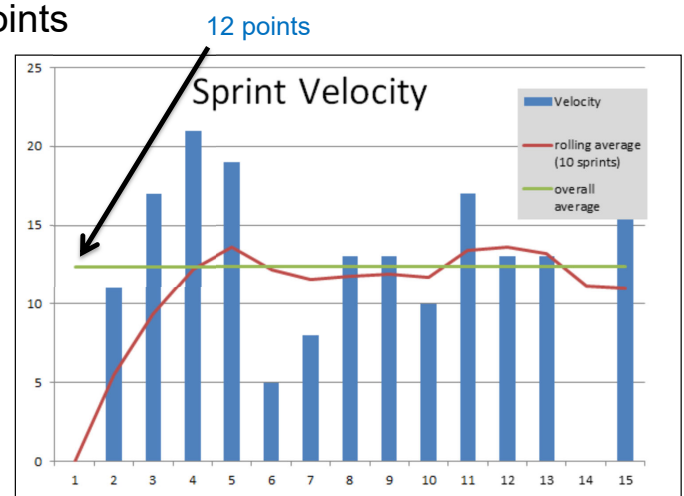
Scenario:

- In one iteration, our team delivers 3 user stories
- The sum of all the story points equals 20
- **Our velocity is then 20**
- In the next iteration, our team delivers 30 story points
- Our average velocity is (20 SP + 30 SP) divided by 2 iterations = **25 SP**

Source: <https://www.scrumalliance.org/community/articles/2014/february/velocity>

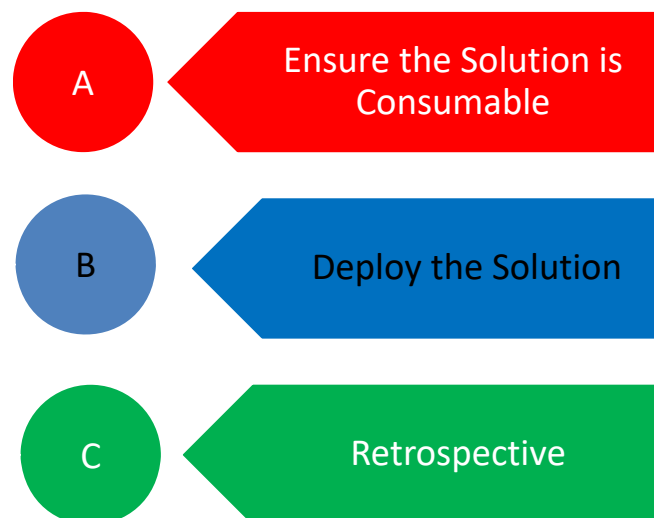
Planning Release Date - Example

- Assume average project velocity = 12 points
- Work Item Stack contains 120 points
- No. of Iterations = $120 / 12 = 10$
- 1 x pre-release iteration for product “hardening” & preparation
- Total no. of iteration = 11
- If each iteration is 2 weeks long, then total estimated duration
= $2 \times 11 \text{ weeks} = \underline{\underline{22 \text{ weeks}}}$

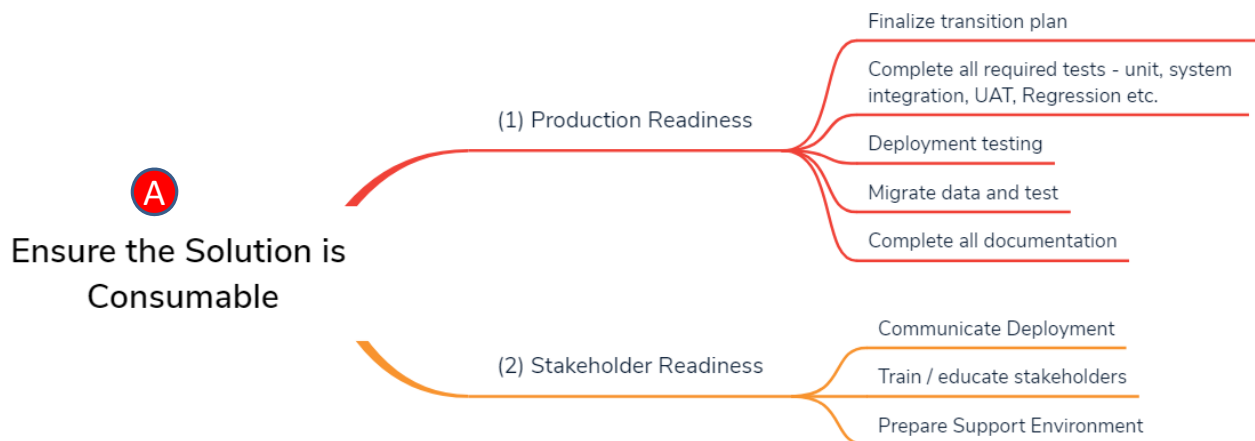


Workshop Part 2

(II) Transition - Objectives



Ensure the Solution is Consumable



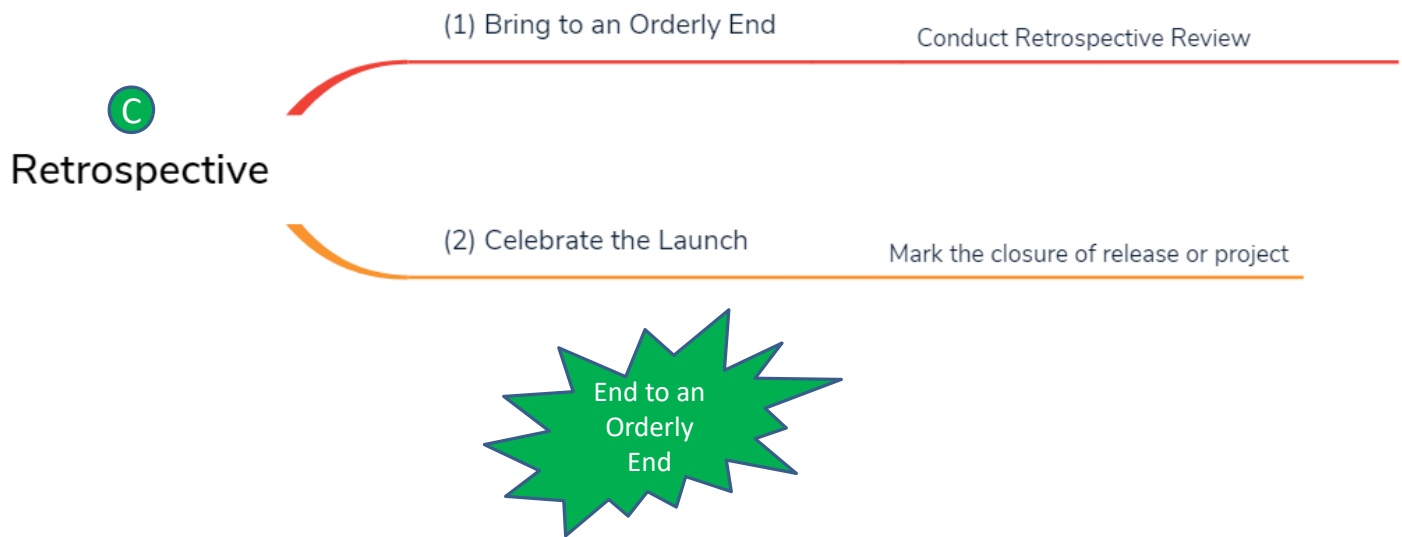
- Important & imperative – must know our stakeholders
- **Is the solution technically ready to be deployed?**
- How do we ensure that there is adequate evidence that the system is reliable and maintainable?
- **Are your stakeholders ready to accept the solution?**

Deploy the Solution



- **What will be required to deploy the solution?**
- How will we determine that the deployment of the solution was successful?
- **Is there any transition roll-out of the solution?**

Retrospective



Retrospective - Some Key Activities

Bring the release to an orderly end by:

- Close off all pending project documentation
- Write final status report (if required)
- For management - assess successes and shortcomings of the work processes
- For both IT and Users - learn from the experience gained
- Re-deploy project resources
- Conduct the Retrospective Review (RR)
- Close off and archive documentation
- Celebrate project completion

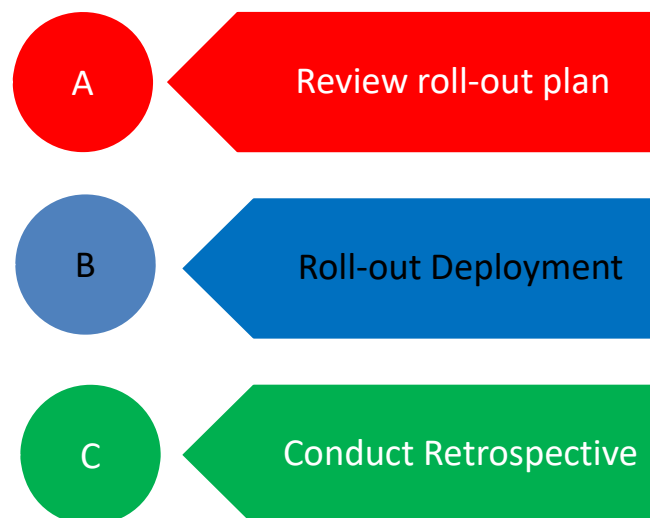


Celebrate The Launch / Completion

- Mark the official closure or release of project
- Leverage the opportunity for formal recognition
- Celebrate event
 - Formal sign-off
 - Commissioning ceremony
 - Product launch event



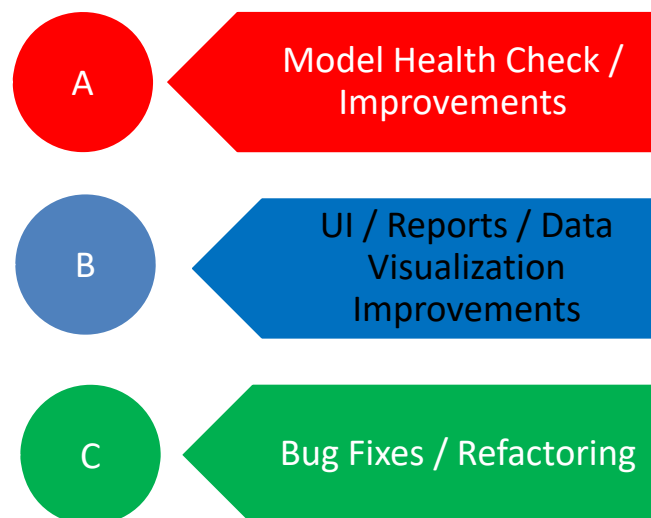
(III) Roll-out - Objectives



Roll-out Considerations

- Must have roll-out plan
 - Phased or big bang approach
- Hardware / software / infrastructure deployment considerations
- Infrastructure & security considerations
- Testing responsibilities & considerations
- Insource or Outsource responsibilities
- Resource deployment
- Retrospective

(IV) Maintenance - Objectives



Maintenance Considerations

- Conduct model health check and look for ways to improve
- Review UI / Data Visualization & Reports
- Fix errors and bug
- Review solution performance and refactor accordingly

THANK YOU 😊

richardt@nus.edu.sg

APPENDIX 1

Verify versus Validate

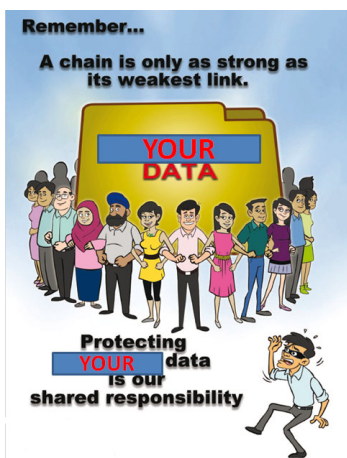
Verify versus Validate

Criteria	Verification	Validation
<i>Definition</i>	The process of evaluating work-products (not the actual final product) of a development phase to determine whether they meet the specified requirements for that phase.	The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements.
<i>Objective</i>	To ensure that the product is being built according to the requirements and design specifications. In other words, to ensure that work products meet their specified requirements.	To ensure that the product actually meets the user's needs, and that the specifications were correct in the first place. In other words, to demonstrate that the product fulfills its intended use when placed in its intended environment.
<i>Question</i>	Are we building the product <i>right</i> ?	Are we building the <i>right</i> product?
<i>Evaluation Items</i>	Plans, Requirement Specs, Design Specs, Code, Test Cases	The actual product/software.
<i>Activities</i>	<ul style="list-style-type: none"> • Reviews • Walkthroughs • Inspections 	<ul style="list-style-type: none"> • Testing

APPENDIX 2

Data Governance & Security Overview

(2) Data Governance & Security



- Leverage on existing IT security policies & artifacts
 - Typical IT Security Concerns
 - Personal Data
 - Basic Data Security Concepts
 - Data Governance – some insights
 - Risk analysis – a balancing act
 - Data classification policy
 - Business rules regarding handling of data & other information assets
 - Codified data / information asset ownership & custody

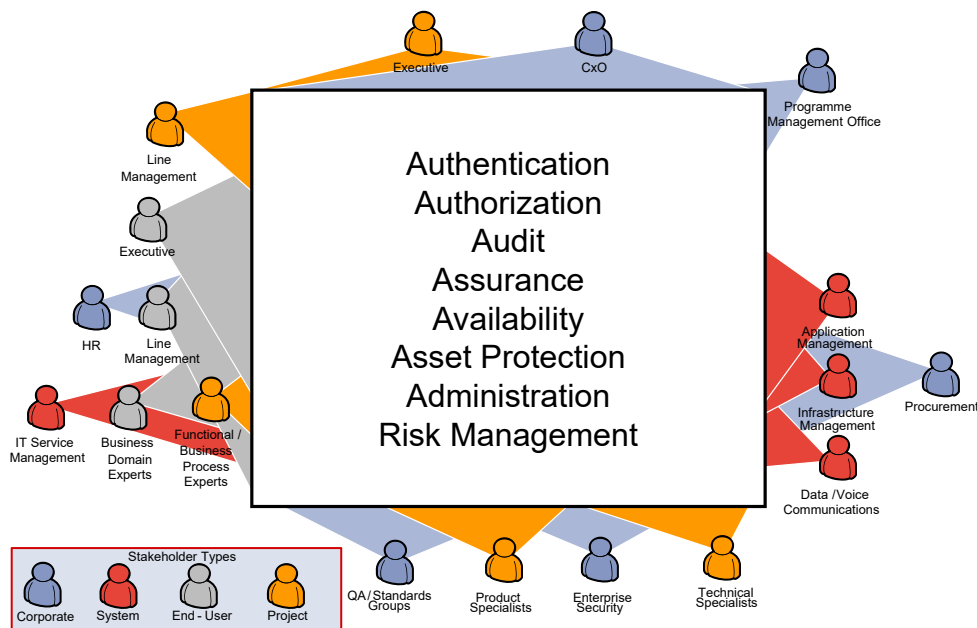
IT Security Policy

- Must have an IT Security policy in place to ensure that data is protected
- Must have minimum security measures required for the protection of information systems as well as the data / information contained and processed by the systems
- Ensure protection against a wide range of threats in order to ensure operations continuity and minimize business damage and maximize return on IT investments
- The systems have different characteristics in the following key areas:
 - Sensitivity of information
 - Criticality to operations of the organization and its Line of Businesses
 - Risk exposure
 - Potential impact in the event of a security breach
- Helps defines security measures to ensure that the data / information assets are protected and consistency in the implementation and practice of security organization-wide

Some IT Protection Strategies

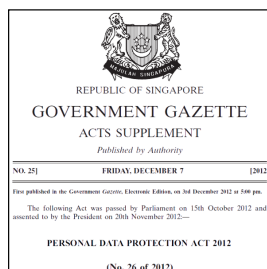
- Multi-layer Security by Design
- 360° Security
- Centralized (and sometime decentralized) management of sites
- Critical Monitoring and real time response
- Audit Checks
- Keep up to date with happenings around in Singapore and the world
- Teamwork
 - Within your organization – auditors; risk management network; infrastructure administrators; application developers...
 - Outside your organization - Industry regulators; government bodies and agencies; industry-key players; Customers
- Perimeter defense
- Physical defense
- Governance – corporate, data, risk, IT...

Typical IT Security Concerns



Personal Data Protection

- Stronger rules on data protection mean people have more control over their personal data
- Businesses will benefit from a level playing field



SGP PDPA Source: <https://www.pdpc.gov.sg/Legislation-and-Guidelines/Personal-Data-Protection-Act-Overview>

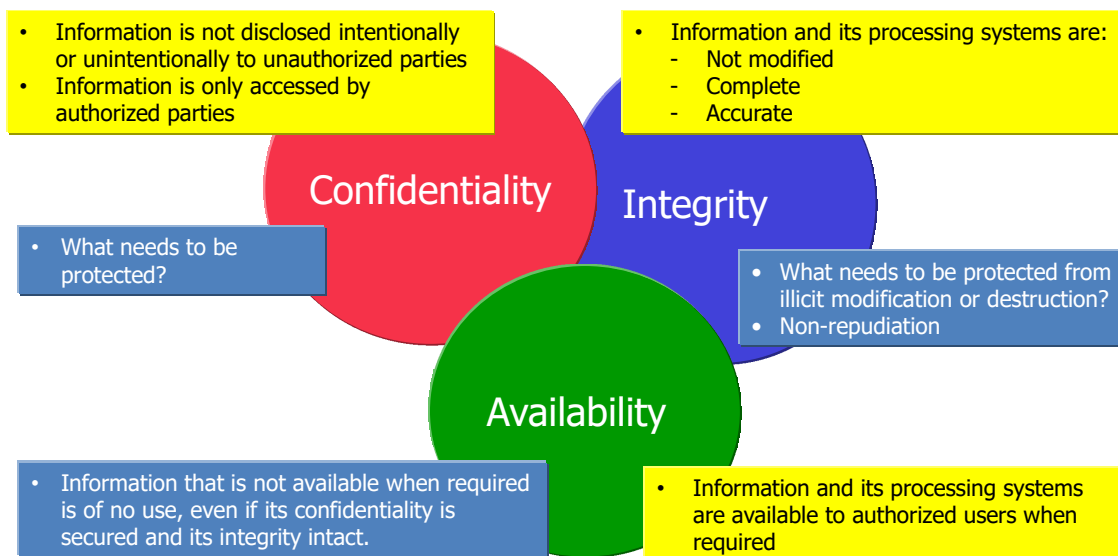
GDPR Source: https://ec.europa.eu/commission/priorities/justice-and-fundamental-rights/data-protection/2018-reform-eu-data-protection-rules_en

Basic Data Security Concepts (1/2)

- Must have definition of what constitutes to be your data
- Any departure from the requirements of policy requires the approval of the relevant authority
- Must have measures in place to protect electronic files; data-in-storage; email transmission; hard-copy protection etc.



Basic Data Security Concepts (2/2)



Data Governance – sharing...

- As a critical organizational asset, data should be shared across the enterprise or organization
- Data sharing can be systematic, routine data sharing or Ad hoc or 'one-off'
- Any request for data shall have a clear and valid purpose
- Data sharing within an organization and with external organisation shall be done with the consent of the Data Owner
- Data sharing is in compliance with the Data Protection Principles.
- Optimize its utilization (include sharing) to authorized and authenticated users over its life-cycle

Refer to the Appendix 3: Some Data use Guidelines

Data From External Providers

- Such data shall be used, stored, managed and administered in **accordance with the terms and conditions of any agreement** relating to the disclosure of such data
- Where no terms and conditions have been specified in relation to data received from external parties, the organization may **use such data for its intended purpose only**
- It **should not pass on such data to any other external parties** without the explicit approval of the originator, and shall exercise reasonable care to protect the confidentiality of such data (where applicable)

Data Security Considerations

- Data Administration & Security (Refer to Appendix 4)
- Data Quality
- Data Classification
- Data Use
- Data Sharing
- Data from External Providers
- Data Standards
- Data Storage
- Data Preservation and Disposal

APPENDIX 3

Some Data Usage Guidelines

Guidelines for Data Use (1/2)

- Refers to the treatment and handling of your enterprise data by Data Users who have been granted access to your enterprise Data
 - Must use in context for the purpose of conducting your enterprise-related business or matters within their scope of work and duty
 - Must comply with all applicable protection and control procedures for your enterprise Data to which they have been granted the right to view, copy, download or otherwise access and/or use

Guidelines for Data Use (2/2)

- Respect the data confidentiality
- Abide by applicable laws and/or enterprise's statutes, regulations and policies
- Do not use, copy, publish, store or transmit enterprise Data in violation of copyright laws
- Do not access and / or use of your enterprise Data for personal gain or profit, or to satisfy personal curiosity
- Data Sharing guidelines is critical
- Protection of Data

APPENDIX 4: Data Administration & Security

Data Administration & Security: Sample Roles & Responsibilities (1/7)

- **Data Owner**
 - Organization is the data owner
 - Individual units or departments may have stewardship responsibilities
- **Data Stewards**
 - Senior staff with planning & policy level responsibilities within their functional areas
 - Establish data definitions & ensuring they are kept current
 - Responsible for data classification
 - Ensure level of data quality is kept acceptable to all
 - Responsible for establishing & controlling data access
 - Work with relevant parties to ensure adequate & appropriate system security requirements for the data
 - Responsible for defining criteria for archiving and data retention to meet legal & business-driven retention requirements

Data Administration & Security: Sample Roles & Responsibilities (2/7)

- **Data Managers**
 - Have operational level responsibility for implementing policies & procedures established by Data Stewards
 - Responsible to ensure that the right users have access to the right data
 - Conversely, responsible to ensure unauthorized access or manipulation does not occur
- **System Owners**
 - Responsible to collate, moderate and integrate business requirements for the development of information systems to meet the objectives and strategic goals of the organization
 - Ensures the “ilities” of the data including usability, reliability, availability etc.

Data Administration & Security: Sample Roles & Responsibilities (3/7)

- **Data Administrator / Architect / Strategist**
 - Coordinates all data management activities within the organization & encourages data sharing
 - Responsible for the overall data architecture, data models for the data repositories
 - Facilitates & coordinates the naming and defining of the data in consultation with the Data Managers & Application Developers
 - Assist & work with the Data Stewards to develop, promote & communicate organizational level policies procedures, standards & guidelines for the management & administration of the data

Data Administration & Security: Sample Roles & Responsibilities (4/7)

- Database Administrator
 - Ensures the “structural” integrity of the physical databases to ensure alignment with the data / information architecture, physical security, recoverability and performance of the databases to meet the multiple needs
 - Responsible to implement the data models and other database maintenance activities

Data Administration & Security: Sample Roles & Responsibilities (5/7)

- Application Developers
 - Responsible for application system development, enhancement and support to meet business requirements and functional specifications of the organization as determined by and agreed with System Owners.
 - Also responsible for the storage and security of the data by ensuring that access is controlled and is available only to authorized users of the data
 - Ensure that access to manipulate data is carried out Application Developers should not have “update” access to production University Data, and not be permitted to update University Data directly via the backend production database to minimise any compromise of data integrity or risk of data loss.

Data Administration & Security: Sample Roles & Responsibilities (6/7)

- **Data Users – subscribers**
 - All persons who have been granted access to organization's Data
 - May include and contractors, consultants, vendors, and contract workers (including their employees, agents and other authorized representatives) and any other person hired by the organization

Data Administration & Security: Sample Roles & Responsibilities (7/7)

- **IT Security Group**
 - Responsible to maintain IT security policies, procedures, standards and guidelines for the organization and reviewing them on a regular basis;
 - Determine and support proper methodologies and processes for IT security
 - Educate users on good IT security practices including formalizing and implementing security awareness programs to raise user awareness and understanding of basic IT security concepts and organization's policies, procedures, standards and guidelines on the same
 - Test & evaluate IT security technologies, systems or applications for deployment within the organization
 - Monitoring & alerting the organization's community of system vulnerabilities documented by product vendors and external IT security organizations and
 - Investigate & handle any IT security incidents or violations