



Smart Scrum

Session 2

Version 1.0

December 2019

CitiusTech has prepared the content contained in this document based on information and knowledge that it reasonably believes to be reliable. Any recipient may rely on the contents of this document at its own risk and CitiusTech shall not be responsible for any error and/or omission in the preparation of this document. The use of any third party reference should not be regarded as an indication of an endorsement, an affiliation or the existence of any other kind of relationship between CitiusTech and such third party.

Agenda

- **Scrum Roles**
- Scrum Ceremonies
- Scrum Artifacts

Scrum Framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily Scrum Meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Scrum Framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily Scrum Meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Product Owner

- Gather customer requirements
- Demonstrate the solution to customers
- Prioritize tasks and create Sprints
- Negotiates priorities, scope, funding, and schedule
- Maintain clear product backlog
- Communicate team status
- Organize milestone reviews
- Educate customers in the development process
- Define and announce releases



Scrum Master

- Helping the Product Owner maintain the product backlog
- Make sure that the needed work is well understood by the development team
- Deciphering User Stories to the development team
- Making sure the development team knows when to mark the task complete
- Coaching the development team
- Promoting self-organization within the team
- Removing any obstacles in the path of the development team
- Facilitating team events to ensure regular progress
- Educating customers on the product and Scrum principles



Team

- Builds the product iteratively and in increments
- Is truly “cross-functional”
- Includes all the expertise necessary to deliver the potentially shippable product each Sprint
- Self-organizing team, with a very high degree of autonomy and accountability.
- Decides how many items to build in a Sprint, and how best to accomplish that goal.
- Team owns the collective responsibility of developing, testing and releasing the Product increment.



Roles and Responsibilities - Game



Agenda

- Scrum Roles
- **Scrum Ceremonies**
- Scrum Artifacts

Scrum Framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

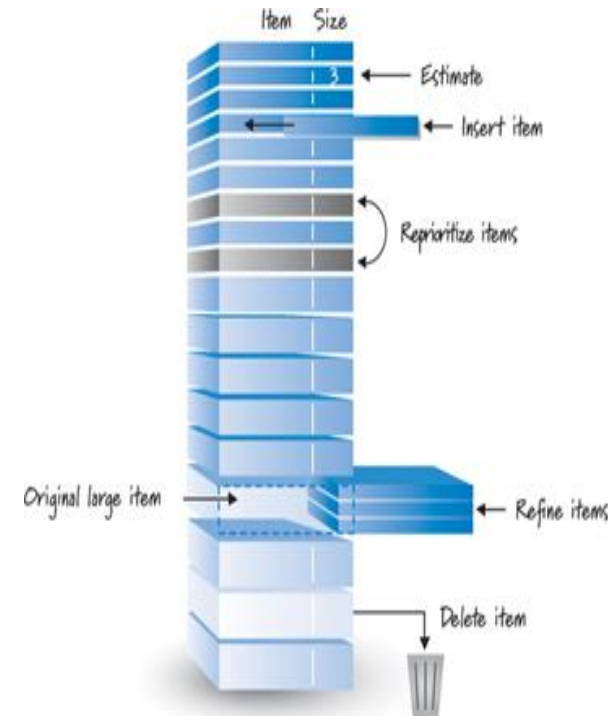
- Sprint planning
- Sprint review
- Sprint retrospective
- Daily Scrum Meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Sprint Planning Meeting

- Conducted at start of Sprint
- Business requirement are presented by PO in form of product backlog
- Team selects items from the product backlog they can commit to completing
- Sprint backlog is created, and stories are selected for the spring
- Sprint Planning contains two activities
 - Story selection
 - Plan confirmation
- Team, Product Owner select backlog for Sprint
 - Product Owner Prioritizes on the importance
 - Team estimates and commits



As a vacation planner, I want to see photos of the hotels.

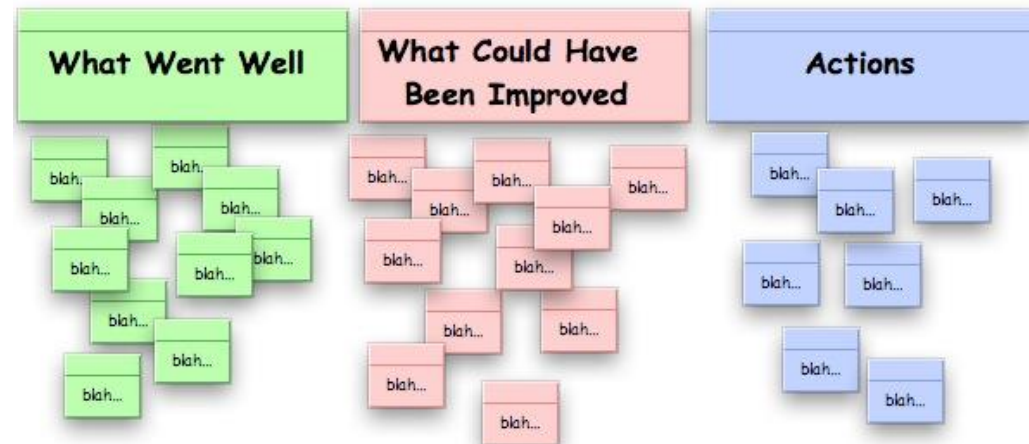
Code the middle tier (8 hours)
Code the user interface (4)
Write test fixtures (4)
Code the foo class (6)
Update performance tests (4)

Sprint Review

- What is Sprint Review?
 - Team presents what it accomplished during the sprint (show and tell)
 - Demo and Answer questions
 - Discuss the next steps/Take feedback
 - In case any changes or new request, Product owner (PO) note and updates the product backlog as required.
 - Product owner is final decision maker on acceptance
- When it is done?
 - Conducted immediately on end of sprint
 - Norms are 1 Hour per Sprint
- How it is conducted?
 - Informal, no prep, show and tell, 2-hour prep time rule, No slides, Whole team participates
 - Scrum master facilitate the environment.
- Who attends?
 - PO, SM, SME, BO, team members
 - All charts, backlog needs to be visible during the review

Sprint Retrospective

- What is Retrospective?
 - Review process, tools, team etc.
 - Periodically take a look at what is and is not working
 - Create improvement plans
 - Lessons learnt
- When it is done?
 - After every sprint
 - Norms are next day of sprint review
- How is conducted?
 - 1 hour per sprint
 - Formal, no prep
- Who attends?
 - Team only - PO, SM, team
 - SM is responsible for conducting Sprint Retrospective
 - Owns the Action item and follow ups
- Where should it be held?
 - Team workspace (not necessary)/ Meeting room
 - Natural break, EOD



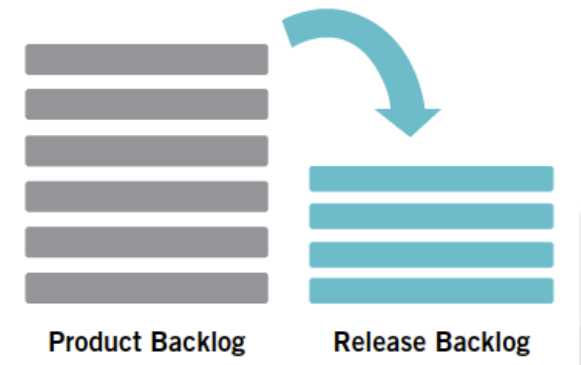
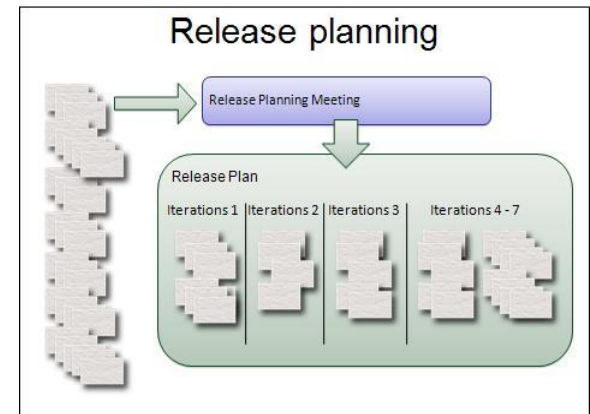
Daily Scrum

- First thing in the day, all team members requires to attend Daily Scrum
- Typically about 15 minutes in duration
 - Discuss the sprint
 - Consistent time and place
 - No discussions on finding solutions to problems
 - Not a status meeting
 - All team member are mandatory
 - SM, PO participate as equal
- Round robin, each team member answers three core questions
 - What did I accomplished yesterday?
 - What will be accomplished today ?
 - What does the other team member need to know about?
 - Obstacles preventing progress, if any?
 - Collaboration with other, if any?
 - Important discovery and learning , if any?



Release Planning

- Typically about 2 to 3 days
- Cross over from Product vision to scrum process
- Taking inputs from team and stakeholders and if needed revise the Release Vision
- Product Owner segments the product into Release1, Release2, etc.
- Decide the major milestones
- Typically a release equates to the implementation of a few top priority features.
- Medium Term Target
 - Foreseeing 3-4 months or 6-8 sprints
 - Understand what is the business value provided in the release
- Guide to selection of stories and prioritization of stories
- It keeps the team focused
- Release strategy should be based on Product Vision



Note: Release Planning is generally outside the SCRUM activities.

Exercise: Input / Output of Scrum Ceremonies

- There are 4 events (or ceremonies) in Scrum.
 - Sprint planning
 - Sprint review
 - Sprint retrospective
 - Daily scrum meeting

- In your teams come up with the input/ output for the above events

Agenda

- Scrum Roles
- Scrum Ceremonies
- **Scrum Artifacts**

Scrum Framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

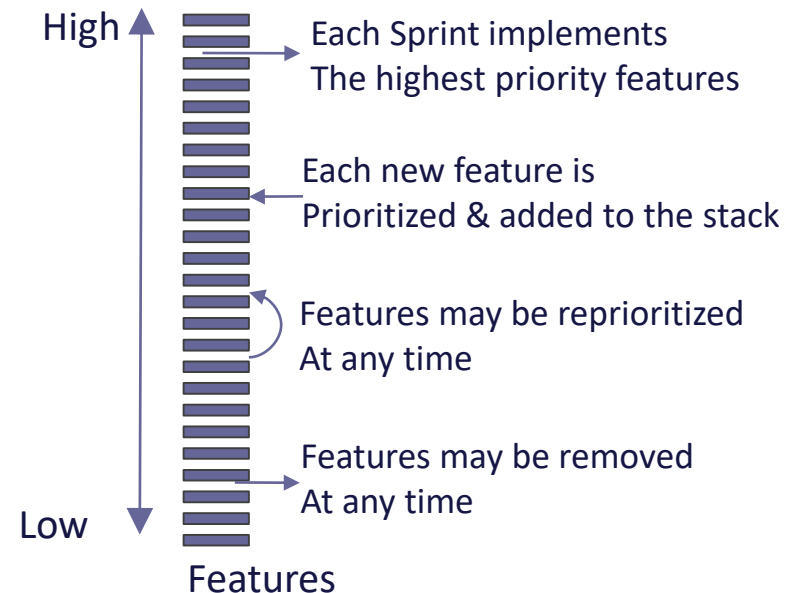
- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Product Backlog

- This is an ordered list of ideas for the product, which can come from the product owner, team members, or stakeholders.
- A description and estimate of effort complement each product backlog item.
- The product backlog is ordered to maximize the value delivered by the Scrum team.
- The development teams work comes from the product backlog, and nowhere else.
- Every feature, enhancement, bug fix, documentation requirement, every bit of work the team does comes from a product backlog item.
- Typically it begins short and becomes longer and more defined as time goes on.
- Though the product owner is responsible for maintaining the product backlog, the development team helps produce and update it.

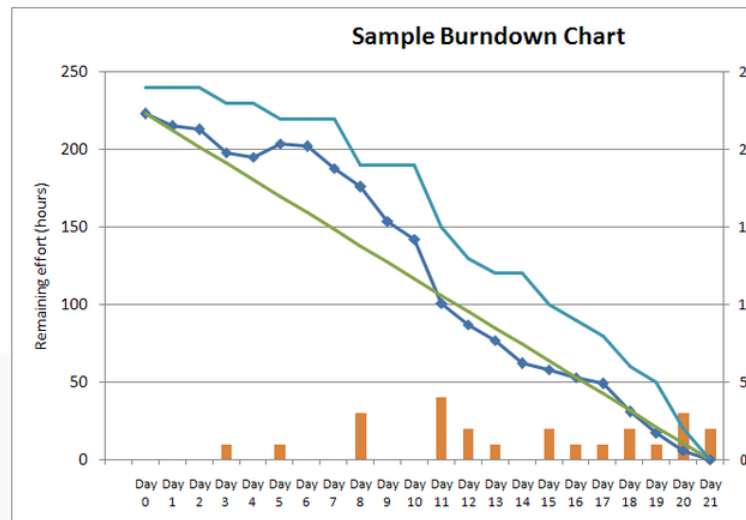


Sprint Backlog

- The sprint backlog is the list of refined product backlog items chosen for development in the current sprint, together with the teams plan for accomplishing the work.
- It reflects the teams forecast of what work can be completed.
- Once the sprint backlog is established, the development team begins work on the new product increment.
- Any team member can add, delete or change the sprint backlog
- Work for the sprint emerges
- If work is unclear, define a sprint backlog item with a larger amount of time and break it down later
- Update work remaining as more becomes known

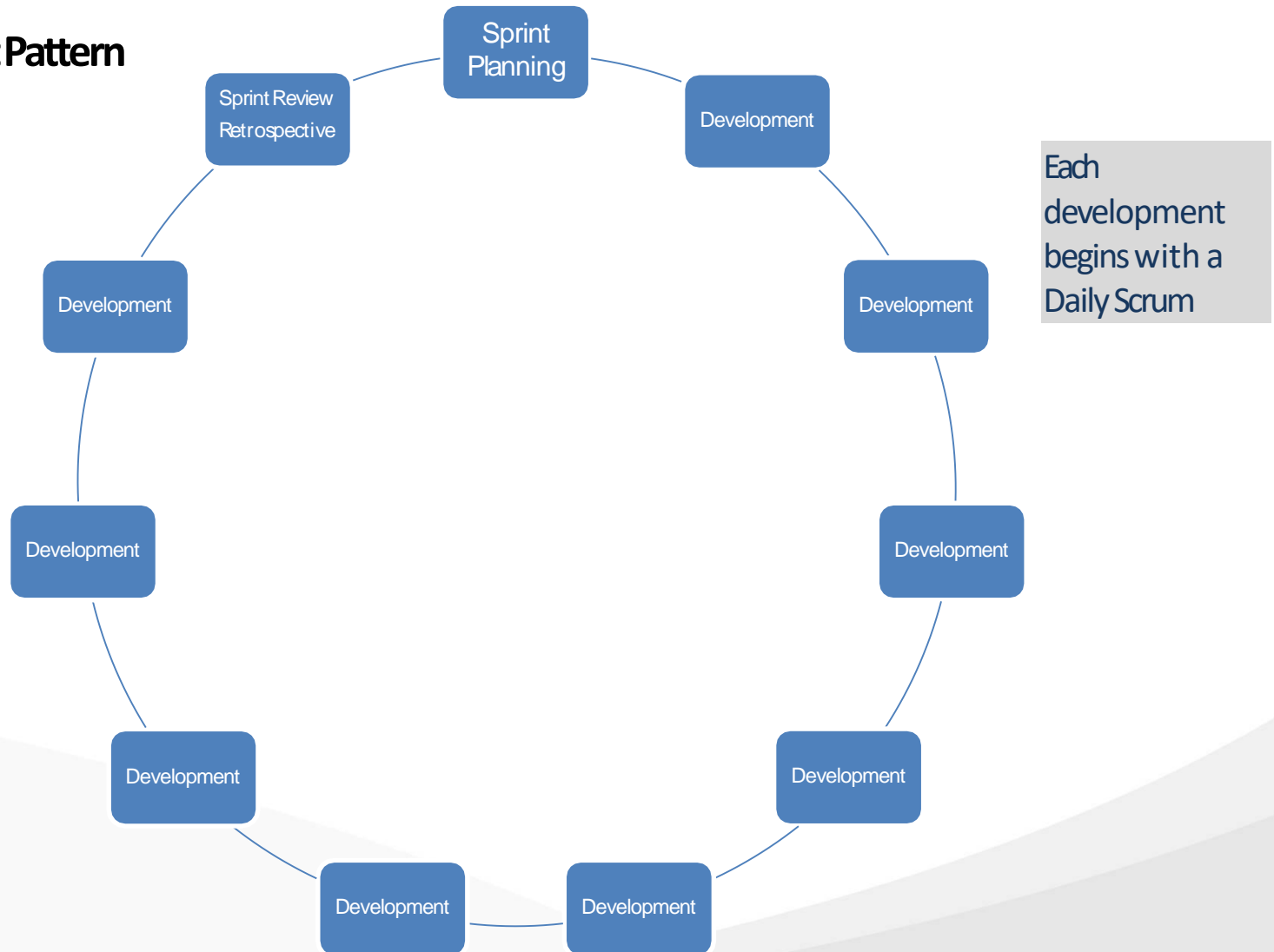
Burndown Charts

- The burndown is a chart that shows how quickly as a team, are burning through your customer's user stories. It shows the total effort against the amount of work we deliver each iteration.
- We can see the total effort on the left, our team velocity on the right. But look what else this simple graphs gives us.
- Burndown's forte:
 - Work done on each iteration
 - Work remaining
 - Work done so far
 - When we can expect to be done



Sprint Timeline

Two Week Sprint Pattern



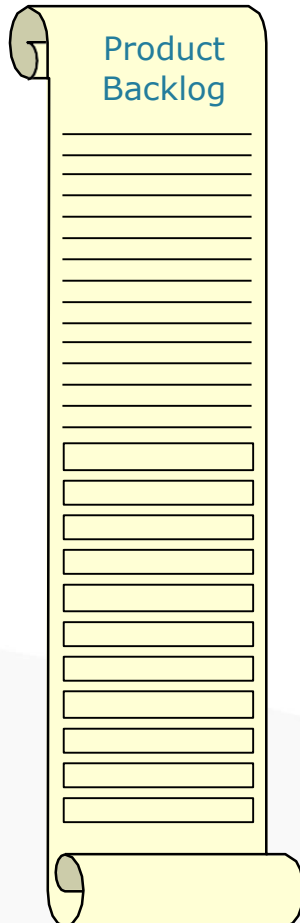
User Stories

- A user story is a tool used in Agile software development to capture a description of a software feature from an end-user perspective. The user story describes the type of user, what they want and why. A user story helps to create a simplified description of a requirement.

User Story		
2	As a <User>	Who?
3	I need to <do something>	What?
1	So that <a result is achieve>	Why?
Defination of Done		
.....		
Business Value		
.....		
Priority		

- **Types of Stories**
 - Epic – Collection or Categorization of related User Stories
 - Technical Stories or Spikes – Related to configurations, POCs, Infra Setup etc

User Story Readiness Guidelines



Independent

Can be delivered independently?



Negotiable

Descriptive enough to support team debate and conversation?



Valuable

Delivers customer or business- visible benefit?



Estimable

Clear enough that team can estimate?



Small

Divided into small enough blocks to complete within Sprint?



Testable

Clear acceptance criteria to know when it is "good enough?"

User Story Prioritization

- There are several ways to prioritize the requirements in the backlog, one of them is **MoSCoW**
 - **M: Must Have.** Describes a requirement that must be satisfied in the final solution for the solution to be considered a success.
 - **S: Should Have.** Represents a high-priority item that should be included in the solution if it is possible. This is often a critical requirement but one which can be satisfied in other ways if absolutely necessary.
 - **C: Could Have.** Describes a requirement which is considered desirable but not necessary. This will be included if time and resources permit.
 - **W: Won't Have.** Represents a requirement that stakeholders have agreed will not be implemented in a given release but may be considered in the future.

Acceptance Criteria

- Acceptance criteria define the boundaries of a user story and are used to confirm when a story is completed and working as intended.
- Acceptance Criteria are a set of statements, each with a clear pass/fail result, that specify both functional and non-functional requirements, and are applicable at the Epic, Feature, and Story Level.
- What are Acceptance Criteria used for?
 - To define boundaries
 - To reach consensus
 - To serve as a basis for tests
- Acceptance criteria are usually initiated by Product Owner but other team members can also participate in defining the acceptance criteria for each story during Backlog Grooming
- Example for Password for a Login Module -
 - Must be at least 8 characters and no more than 12
 - Must contain only alpha numeric
 - Must contain at least one digit
 - Must contain at least one character
 - Etc. (There would be more such criteria)

The Definition of “Done”

- How does the Development Team know when it has met a requirement?
- Scrum requires that every increment be “potentially shippable”
 - This means gaining agreement about what it means to be “done”
 - The Product Owner may decide to deploy it immediately even if the Sprint is not at the end of a scheduled release
- Different people have different perspectives
 - A programmer might believe “done” is when the code is complete
 - A tester is “done” when unit and integration testing is complete
 - A customer might regard “done” as
 - Installation, deployment of the system
 - Together with user documentation
- In Scrum, “done” is an agreement between the Product Owner and the Development Team

Exercise: Writing User Story

Case Study : Safe Flight Airlines

- Safe Flight Airlines is a regional airline that transports over 7 million customers each year between 40 cities and three countries
- Safe Flight's consistent, profitable growth has resulted from the airline's reputation for outstanding customer service
- Safe Flight was one of the first airlines to sell tickets online, but has fallen behind recent innovations in the industry
- Safe Flight has identified their outmoded check-in process as a way to reduce costs and increase customer satisfaction to increase yield

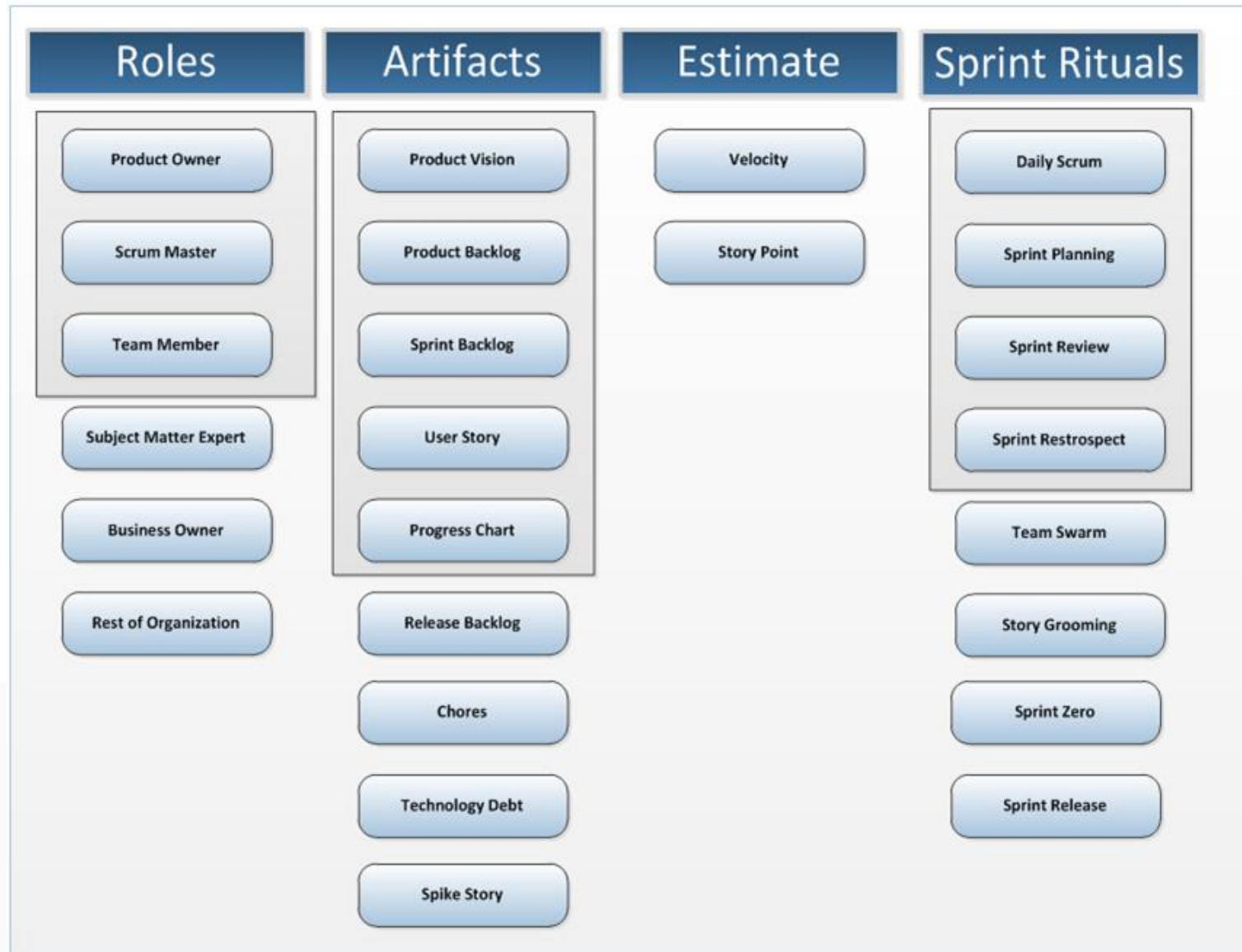
In your teams come up with 5 stories for improving the Safe Flight Airline's Check-In process where it can improve the overall experience for the customer

Scrum Tools – A Glance...

Licensed Tool	Open Source
JIRA	Icescrum
Rally	Agilefant
Team Foundation Server – TFS	
Version One	
Mingle	

Reference for Scrum tools : <http://agiletools.info/>

Summarize



THANK YOU



CitiusTech
Markets



CitiusTech
Services



CitiusTech
Platforms



Accelerating
Innovation

CitiusTech Contacts

Email ct-univerct@citius-tech.com

www.citius-tech.com