

SCRUM

SCRUM IS AN AGILE PROCESS THAT ALLOWS US TO FOCUS ON DELIVERING THE HIGHEST BUSINESS VALUE IN THE SHORTEST TIME.

IT IS BASED ON:

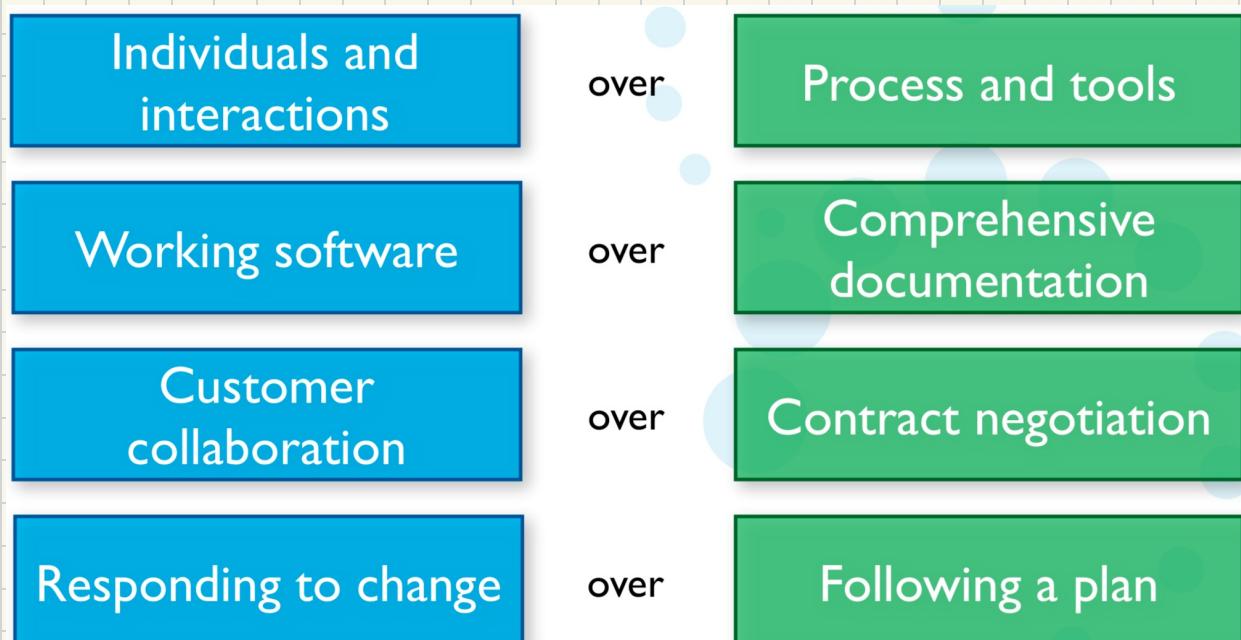
- SHORT AND FREQUENT ITERATIONS (2-4 WEEKS) CALLED SPRINTS.
- A CONTINUOUS FOCUS ON THE HIGHEST PRIORITY FEATURES.
- SELF-ORGANIZING TEAMS THAT DEFINE HOW WORK GETS DONE
- THE ABILITY TO RELEASE WORKING SW AT THE END OF EACH SPRINT.



CHARACTERISTICS

- **SELF-ORGANIZING TEAMS:** EACH TEAM INDEPENDENTLY DECIDES HOW TO COMPLETE THE WORK.
- **ITERATIVE SPRINTS:** EACH SPRINT PRODUCES A WORKING INCREMENT OF THE PRODUCT (NO CHANGES HERE).
- **PRODUCT BACKLOG:** REQUIREMENTS ARE COLLECTED AS A LIST OF PRIORITIZED FEATURES.
- **GENERATIVE RULES:** CREATE AN AGILE ENVIRONMENT WITHOUT IMPOSING SPECIFIC TECHNICAL PRACTICES.

THE AGILE MANIFESTO



SCRUM FRAMEWORK

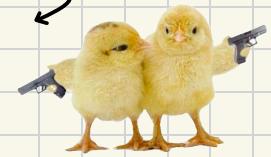
• ROLES:

- **PRODUCT OWNER:** DEFINES AND PRIORITIZES FEATURES, ACCEPTS OR REJECTS THE SPRINT RESULTS.
- **SCRUM MASTER:** FACILITATES THE PROCESS, REMOVES OBSTACLES AND PROTECTS THE TEAM FROM EXTERNAL INTERFERENCE.
- **TEAM:** CONSISTING OF 5-9 MEMBERS, CROSS-FUNCTIONAL AND SELF-ORGANIZING.

• CEREMONIES:

- **SPRINT PLANNING:** SPRING ACTIVITY PLANNING
- **DAILY SCRUM:** DAILY 15 MINUTE MEETING TO SYNCHRONIZE THE TEAM. EACH MEMBER ANSWERS 3 QUESTIONS →

WHAT DID YOU DO YESTERDAY?
WHAT WILL YOU DO TODAY?
IS ANYTHING IN YOUR WAY?



- **SPRINT REVIEW:** PRESENTATION OF COMPLETED WORK. THE TEAM PRESENTS PROGRESS INFORMALLY (DEMONSTRATIONS, DISCUSSIONS), WITHOUT SLIDES. INVOLVES ALL INTERESTED STAKEHOLDERS.
- **SPRINT RETROSPECTIVE:** AT THE END OF EACH SPRINT THERE IS AN EVALUATION OF WHAT WORKED AND WHAT TO IMPROVE. THE TEAM REFLECTS ON:

WHAT TO START DOING
WHAT TO STOP DOING
WHAT TO CONTINUE DOING

• ARTIFACTS:

- **PRODUCT BACKLOG:** PRIORITIZED LIST OF ALL FEATURES. ELEMENTS ARE OFTEN EXPRESSED AS USER STORIES, i.e. SIMPLE DESCRIPTIONS THAT CONNECT USER, GOAL AND VALUE.
- **SPRINT BACKLOG:** CONTAINS SPECIFIC TASKS AND THE ESTIMATED HOURS TO COMPLETE THEM.

Backlog item	Estimate
Allow a guest to make a reservation	3
As a guest, I want to cancel a reservation.	5
As a guest, I want to change the dates of a reservation.	3
As a hotel employee, I can run RevPAR reports (revenue-per-available-room)	8
Improve exception handling	8
...	30
...	50

Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	4	
Test the middle tier	8	16	16	11	8
Write online help	12				
Write the foo class	8	8	8	8	8
Add error logging				8	4

- **SPRINT BURNDOWN CHART:** A DISPLAY OF WHAT WORK HAS BEEN COMPLETED AND WHAT IS LEFT TO COMPLETE.

Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	7	
Test the middle tier	8	16	16	11	8
Write online help	12				



FOR LARGER PROJECTS, THERE IS THE CONCEPT OF "SCRUM OF SCRUM", WHERE MULTIPLE TEAMS COLLABORATE BY COORDINATING AT A HIERARCHICAL LEVEL.

USER STORIES

USER STORIES ARE CONCISE AND SIMPLE DESCRIPTIONS OF HOW AN APPLICATION IS USED. THEY SERVE TO CLARIFY USER REQUIREMENTS BEFORE AND DURING DEVELOPMENT, REDUCING MISUNDERSTANDINGS.

THEY ARE CENTRAL TO BEHAVIOR-DRIVEN DESIGN (BDD), WHICH FOCUSES ON THE BEHAVIOR OF THE APPLICATION RATHER THAN ITS IMPLEMENTATION.

USER STORIES ARE FORMULATED IN 1-3 SENTENCES IN EVERYDAY LANGUAGE.

AS A KIND OF STAKEHOLDER,
SO THAT I CAN ACHIEVE SOME GOAL,
I WANT TO DO SOME TASK.

THEY USE THE CONNEXTRA FORMAT →

EACH USER STORY IS DESIGNED TO BE VERIFIED AS AN ACCEPTANCE TEST BEFORE WRITING CODE.

PRODUCT BACKLOG MANAGEMENT

A REAL SYSTEM CAN HAVE HUNDREDS OF USER STORIES. THESE ARE COLLECTED IN THE PRODUCT BACKLOG, WHICH CONTAINS ALL THE STORIES NOT YET COMPLETED.

THE BACKLOG IS PRIORITIZED, ALLOWING TO ASSIGN THE MOST IMPORTANT FEATURES TO THE FIRST RELEASES.

SPIKE

A SPIKE IS A TIME-LIMITED EXPLORATION INTO A SPECIFIC TECHNIQUE OR PROBLEM. THE CODE PRODUCED DURING A SPIKE IS DISCARDED ONCE THE SOLUTION IS FOUND.

MEASURING PRODUCTIVITY

TEAM PRODUCTIVITY CAN BE MEASURED IN TERMS OF STORY POINTS, A SCALE THAT EVALUATES THE COMPLEXITY OF EACH USER STORY:

1 SIMPLE STORY.

2-3 MEDIUM COMPLEXITY.

5+ COMPLEX STORY, TO BE DIVIDED INTO SMALLER STORIES.

STORIES VS LAYERS

- “Dividing work by stories helps all team members understand app & be more confident when changing it”
- “Tracker helped us prioritize features and estimate difficulty”

- “We divided by layers [front-end vs. back-end vs. JavaScript, etc.] and it was hard to coordinate getting features to work”
- “It was hard to estimate if work was divided fairly...not sure if our ability to estimate difficulty improved over time or not”



SMART USER STORIES

SPECIFIC. EACH SCENARIO IS TESTABLE AND HAS KNOWN INPUTS WITH EXPECTED OUTCOMES.

MEASURABLE: THE OBJECTIVES ARE CLEARLY DEFINED.

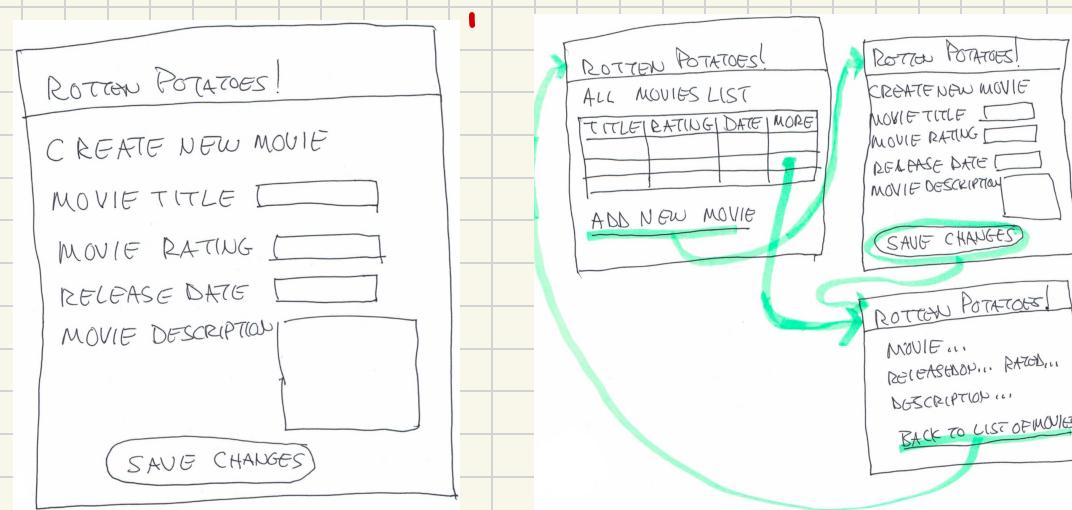
ACHIEVABLE: IMPLEMENTABLE IN A SINGLE ITERATION.

RELEVANT: THEY MUST BRING VALUE TO THE BUSINESS.

TIME BOXED: EVERY STORY MUST HAVE A TIME LIMIT; IF NOT COMPLETED, IT MUST BE DIVIDED OR RESCHEDULED.

LO-FI UI SKETCHES AND STORYBOARD

FOR SaaS APPLICATION, IT'S ESSENTIAL TO DESIGN A UI INVOLVING THE CUSTOMER.
LO-FI¹ PROTOTYPES (UI SKETCHES) AND STORYBOARDS² HELP REPRESENT INTERACTIVITY WITHOUT INVESTING TOO MUCH TIME IN DETAILED PROTOTYPES.



WORKING WITH THE CUSTOMER

- “Lo-fi and storyboards really helpful in working with customer”
- “Frequent customer feedback is essential”
- “What we thought would be cool is not what customer cared about”



- “We did hi-fi prototypes, and invested a lot of time only to realize customer didn’t like it”
- “Never realized how challenging to get from customer description to technical plan”

