Week 4 Day 5

Scrum

Agile: Scrum Methodology

REVATURE

Scrum is the simplest implementation of the Agile framework

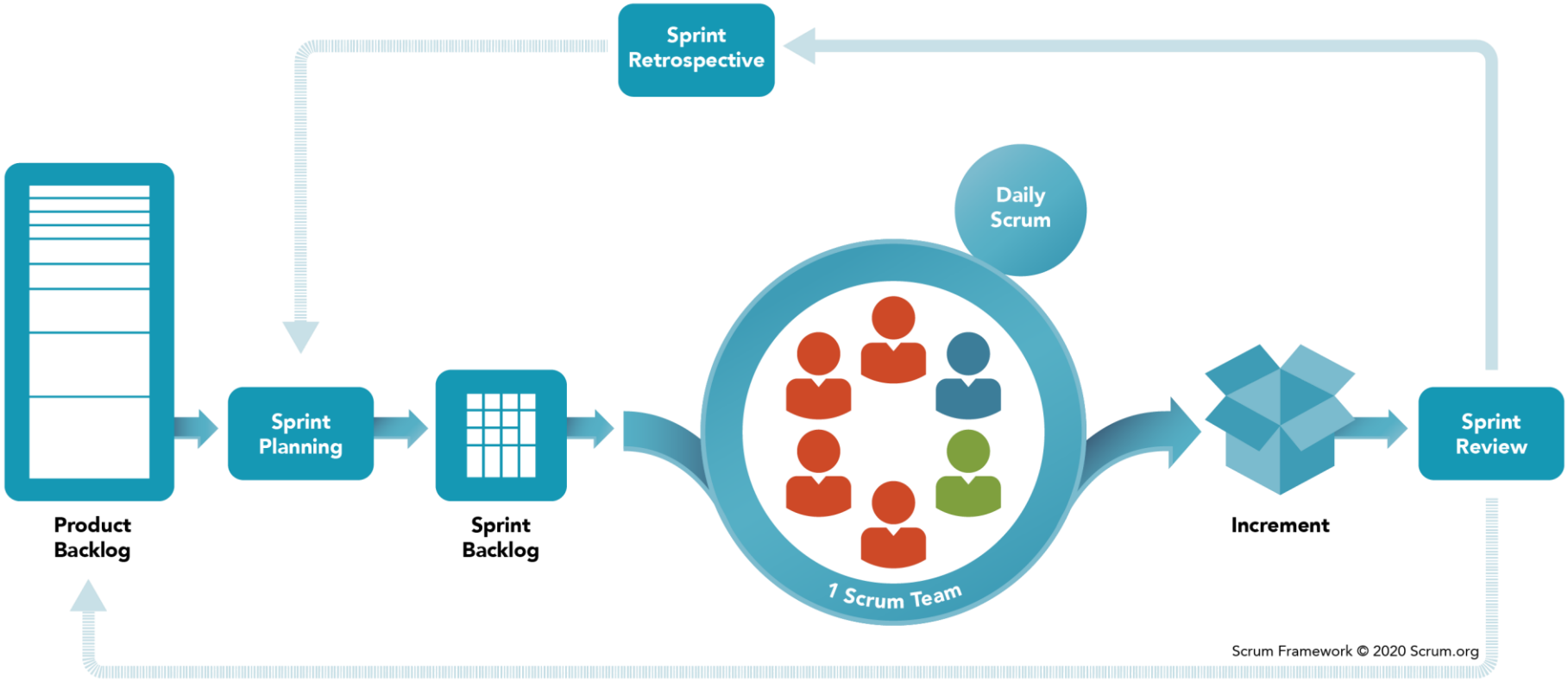


Image source: https://www.scrum.org/resources/what-is-scrum

Scrum: Artifacts

REVATURE

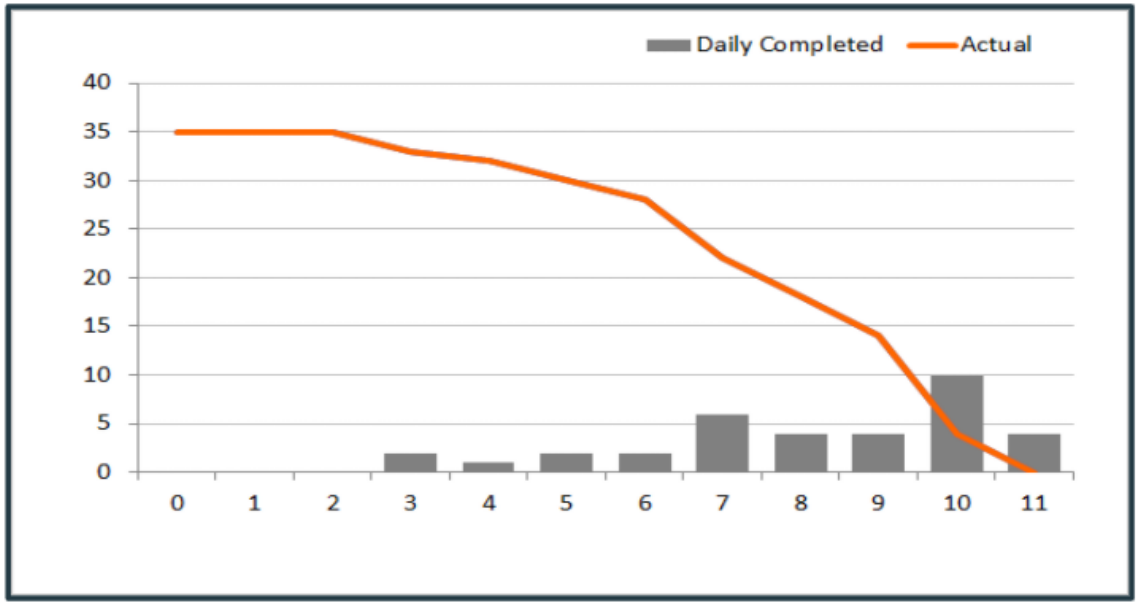
- Product Owner
  - Client/customer
- Scrum Master
  - Team leader
- Project Backlog
  - List of all project requirements
- Sprint backlog
  - List of sprint requirements
- User Story
  - An individual requirement
- Sprint
  - Period in which the team is developing
- Velocity
  - Sum of story points of all user stories complete that week

Scrum: Story Pointing

REVATURE

Assigning Points to each user story

- Used to create burndown charts which show progress
- Assigning points is a team activity
  - Consider the risk, complexity, and repetition of each user story
  - Recommended to use Fibonacci sequence when assigning points



Scrum: Ceremonies

REVATURE

- Spring Planning
  - Devs, leaders, owner
  - Before every sprint
  - Determine scope, goals, and metrics
- Daily Standup
  - Lead by scrum master everyday
  - Quick 15 minutes
  - What you're working on, stuck on, and goals
- Spring Review
  - Anyone and everyone
  - Review what the team accomplished
  - Feedback
- Sprint Retrospective
  - Scrum master reviews metrics, and assesses efficiencies
  - Make improvements

NodeJS

Javascript Runtime Environment

Node.JS

REVATURE

Open source, cross-platform, Javascript run-time environment

- Not a programming language
- Allows us to run JS outside of browser

Node Package Manager

REVATURE

Package manager for JS

- Default manager of the node environment
- Consists of three components
  - Website: discovers packages, sets up profiles, and manages access to packages
  - CLI: runs in the terminal, interacts with npm
  - Registry: public repo for node packagse

package.json

REVATURE

Information/metadata for the project

- Description
- Version
- License information
- Author
- Entry point
- Scripts
- Dependencies

```
{
  "name": "node-example",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  ▷ Debug
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

```

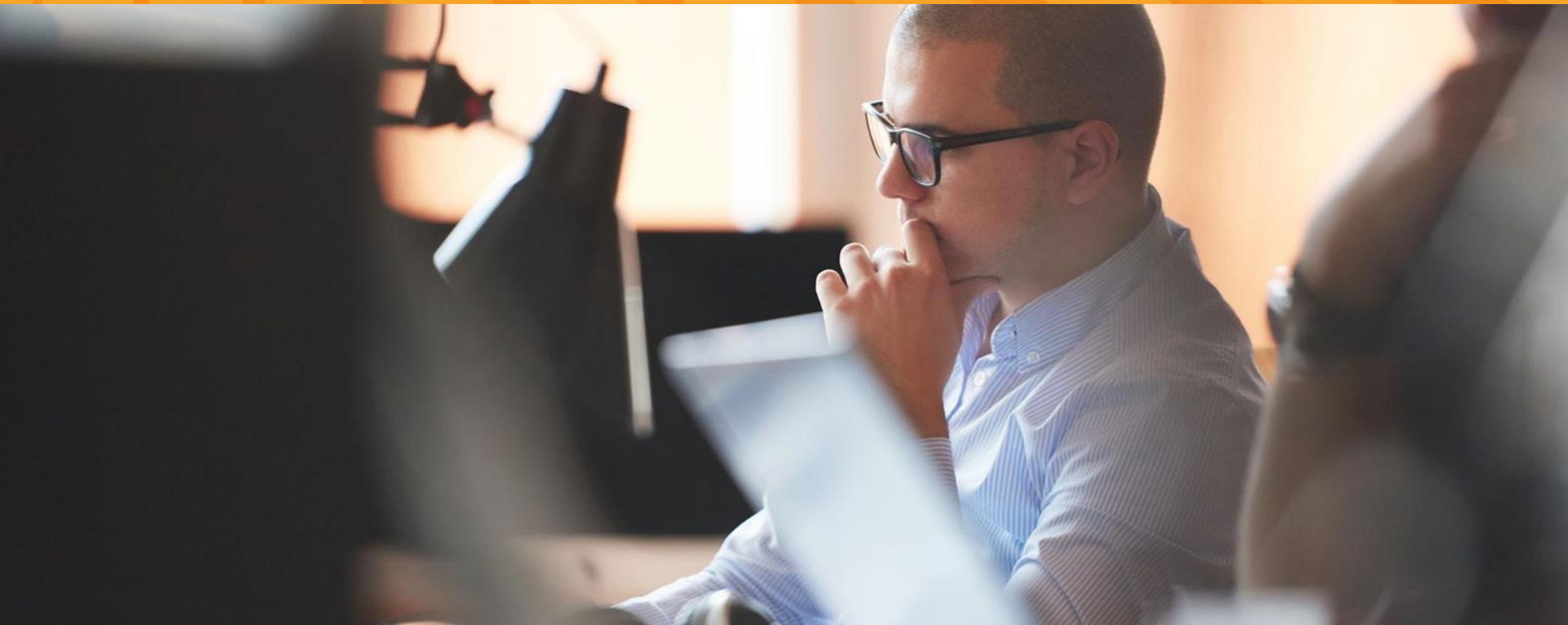
ES6: Modules

REVATURE

- Modules allow developers to export and import javascript code
- Use the `export` keyword to create a module
- User the `import` keyword to import the module
- With NodeJS you must set the “type” property in the package.json to “module” to use it

REVATURE

Install NodeJS



Typescript

Introduction

Typescript

REVATURE

Open-source, object-oriented, typed, superset of Javascript by Microsoft

- All functionality of Javascript
- Additions of classes, interfaces, modules, more
- Portable
- Strong/static typing

Typescript: Compiling and installing

REVATURE

- Install typescript with node package manager
  - npm install –g typescript
- Browser is unable to read typescript directly, we must transpile into javascript
  - Use the command tsc to output the corresponding javascript

Typescript: Data types

REVATURE

Declare the type when declaring a variable:

```
let varname: [type] = value
```

- Boolean
- Number
- String
- Undefined
- Null
- Any
- Void
- Array
- Tuple
- Enum
- Never

Typescript: Classes and Access Modifiers

REVATURE

Typescript adds extra class functionality to JS

- Use of the class and new keywords
- Inheritance with extends keyword
- Access Modifiers
  - Public
  - Private
  - Protected
- readonly modifier
- Getters and setters accessed like properties

Typescript: Interfaces

REVATURE

Create contracts for classes/objects to implement

- Defined with the interface keyword
- Optional properties denoted with ?
- Not actually compiled

```
1 export interface IUser {
2     id?: number;
3     username: string;
4     personId?:number;
5     email?:string;
6 }
```

REVATURE

Typescript DEMO