# Scrum processes

**Scrum Board**

**DSR**

* The DSR is to be sent out at the end of the day
* Always open the dependency in jira and confirm if it is resolved

**MSR(Bi Weekly presentation)**

**Sprint Closure summary**

* To be sent on every bi-weekly Monday.
* Contains list of closed story
* List of story pending to merge to master

**Retrospective meeting**

* Team need to decide the Good and bad part of the sprint
* Voting needs to be done
* Based on voting experiment needs t o be run among team
* Outcome of experiment needs to tracks till the closure
* Each experiment should have responsible person to take the experiment till the closure.

**JIRA**

* Update Jira for all client query and relevant update.

**Confluence(intranet)**

* For documentation

**Code review (stash)**

* For every story, Code should be reviewed by peer and leaders
* Once internal review is approved then client should be added for external review
* RR log needs to be updated for all the list of PRs reviewed In Jira.

**Jenkins build for App109**

* App109 is the integrated environment set up at Turn for CSS Refresh
* All the regression cycles will be run on this environment

**Demos**

* At the end of every sprint, There is a demo setup at client end

**Others**

* You might need conference room booking access and also conference dial-in code

**Goto persons**

* Usability : Surendra, PA : Ben are Technical Lead and co-ordinates with us for any technical challenges
* Sherman Luk is the Engineering Director for CS .  So specifically he makes sure that we all are on track and also in case of any blockers, he needs to be kept informed
* Philip the PMO and does all the co-ordination

**Merge Request**

* Merge request should only contain the intended changes, no refactoring or formatting please

**Email**

* Do not share you password/code/files over skype or email

**Testing**

* Make sure the code is tested in respective browser
* IN CS developer will do the testing in integration environment, developer need to label such ticket as **DevTest**

**Release** :

* Send out what is to be released by end of spring
* Alpha
* CR
* Beta
* GA
* GTM
* <https://intranet.turn.com/display/PMO/4%29.+Release+Milestones+for+Gate+Version+and+Criteria>

**Knowledge Series**:

* We have bi-weekly knowledge series session, where one person present anything new in project, this include
* new important feature introduce in project and worked by team
* New technical enhancement/tool introduce in project
* New version/feature introduce in the technology
* Improving domain knowledge

**Training by weekly**:

* Training happens to upgrade the technical knowledge in the team for used technology

# Hiring Guideline

**Selection criteria**

1. Very good JavaScript & no java knowledge
2. Average JavaScript and knows few JS library (Backborn/knockout/Angular) basic java knowledge.
3. Basic JavaScript , CSS and JQuery knowledge & very good Java with different framework knowledge(this is to check if candidate has capability to lean new thing)

**Note:** in all of above case candidate **must** be ready to work on JavaScript

**General Guide line**

1. Ask Candidate to write the code
2. No definition, More than Knowledge once should ask its purpose and why it was preferred
3. Few Hints are ok to if they come up with good approach
4. Look for get it done attitude
5. Do technical interview but observe the soft skill as well
6. Avoid tricky question or the question whose answer can be found easily on web
7. Observe if candidate choose to write code in paper by himself
8. Check if candidate makes the eye contact to answer the question
9. Observe if candidate ask enough question to clarify the requirement

**Java script**

**Must know**

1. Candidate should be willing to work on java script
2. Objects: how to define object with property and method? Etc.

function Person(firstName, lastName, age, eyeColor) {

    this.firstName = firstName;

    this.lastName = lastName;

    this.age = age;

    this.eyeColor = eyeColor;

    this.changeName = function (name) {

        this.lastName = name;

    }

}

Var employ = new Persion('firstName', 'Last name')

Persion.prototype.getnameByFirstName = function(){

}

Object literal var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};

1. Element selectors (JQuery or purer JavaScript )

document.querySelector(".example")

1. function Literal: How to define, how to call and how to pass as argument

1. Call back function

var friends = ["Mike", "Stacy", "Andy", "Rick"];

friends.forEach(function (eachName, index){

console.log(index + 1 + ". " + eachName); // 1. Mike, 2. Stacy, 3. Andy, 4. Rick​

});

1. Explain 'this' keyword
2. Self-invoking function
3. Closer
4. Difference between setTimeout vs SetInterval
5. Difference between coreJS vs Jquery
6. Prototype : inherited the method
7. Add remove the element from array, : pop, push, delete
8. Difference between == and ===?
9. Namespace
10. JSON structure
11. Any JS library
    1. RequiredJS, AmberJS, Backborn, Knockout or Angular
12. How would you update the title in Single page application in each page
13. How would you get the http response back from service component
14. How would two directive communicate with each other

**Good to know**

1. Module,
2. Call & apply
3. Hoisting( variable/method)
4. Closure
5. task/tool
   1. RequiredJS
   2. Minification
   3. JS hint
   4. Cascading
6. 'use strict'
7. Grunt
8. Bower

**Java**

**Must know**

1. Collection --> contains check, equals check
2. Instance of check, exception check
3. OPPS concept, overriding example
4. Class/abstract class/ interface
5. Create immutable class
6. Concept of thread & concurrent package in JDK 5
7. How to create custom Checked and unchecked exception?
8. When do you catch an exception or let it bubble up to the calling method?
9. Generic in collection
10. Access modifiers
11. Test case writing
12. How do I include static files within a JSP page
13. context initialization parameters
14. Servlet filter

**Good to know**

1. OPPS principle
2. Pattern, Few pattern --> except singleton and factory
3. Race condition
4. Any framework (Hibernate/Scripting etc.)
5. Build tool --> Ant/Maven etc.
6. Write a program which will result in deadlock? How will you fix deadlock in Java?
7. No dead lock, n resource, n thread
8. Restrict only 3 instance to be created
9. REST Security
10. Design the cache which can evict old object, Distributed cache.
11. Explain java Heap and garbage collection
12. Retry on concurrency Error
13. ACID property
14. Optimistic lock
15. GC algorithm e.g. Mark and Sweep
16. Time and Space complexity
17. UML notation

**Advance topic1**

1. Thread
2. Datastructure
3. Security
4. Transaction
5. Caching
6. Time and space complexity

**Project level question**

1. Spring security implementation (custom persmission)
2. Auditing requirement
3. Transaction configuration
4. How Elastic search implementated
5. How to prevent concurrent login

**Test case**

**Must know**

1. In Junit purpose of the setUp() and tearDown() methods?
2. How do I test a method that doesn’t return anything
3. What is the purpose of Mock Objects in unit testing

**Good to know**

1. Difference between mock and spy, when to use what?
2. Assert Behavior

**CSS**

**Must know**

1. Box model --> How to get the width/height of element, what does it considered?
   1. Concept : Margin/Padding/border/content
2. Inline CSS vs External CSS
3. How to override the CSS properties
4. Position/floats
5. Selectors

**Good to know**

1. SAAS/less

**Interview question from turn**

<https://intranet.turn.com/display/ENG/Champaign+Interview+Questions>

**Algorithm**

1. Palindromic Numbers
2. [disarium-number](http://www.guideforschool.com/1929145-java-program-to-check-for-disarium-number/)
3. [prime-number-program-in-java](http://www.javatpoint.com/prime-number-program-in-java)
4. [Strong number](http://www.answers.com/Q/What_is_strong_number_in_C_language) -> sum of factorials of the digits is equal to the given number
5. <http://www.javatpoint.com/directload.jsp?val=86>