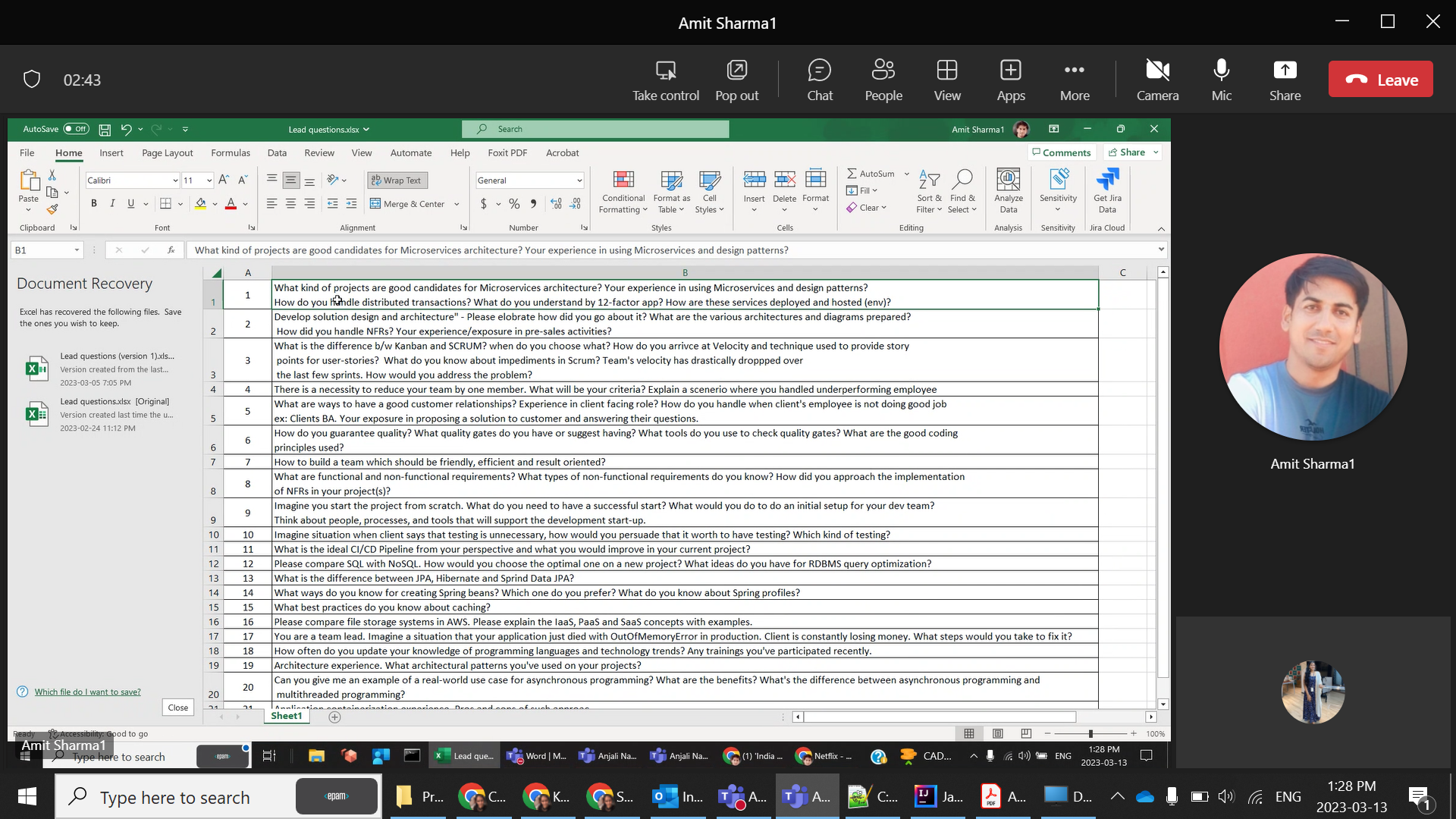
Tech in Grow :



Unit Testing – Mockito & Junit - CORS Cross Origin Resource Sharing   issue how did we fix it ?



Graphical user interface, text, application

Description automatically generated

Clean Code --Practices , Fortify and SONAR . CheckStyle . Code Quality.

**Correctness** - final declaration for required data types - avoid Object creation - Scanner slows down, use InputReader - avoid too many static functions

**Efficiency** - use library functions as much as possible - write freaking jUnit @Test-ing - assertEquals("RESULT", functionToCall())

**Debuggability** - avoid too many global variables - Separate logic from Pre-processing - variable/function pneumonics must make sense

Engx Bootcamp Practices – What are they ?

Set of rules defined in EPAM w.r.t. , Coding standards, Testing ( Unit , Automated) , Code Review , Branching techniques , Continuous Integration.

SOA + Microservices

Loose coupling is an extremely important concept to the understanding, and motivation to create, an SOA designed environment. Without it, one cannot create an SOA designed application. Not only does SOA need a use loose coupling, it also uses what is called an enterprise service bus (ESB) to communicate between the different services

Eg: Dot Net and Java Applications can communicate on cloud technology .

At their heart, these microservices are just a monolithic application’s core functions, and are typically self-contained with complete web capabilities.

We use APIs i.e. used to communicate using HTTP / REST protocols.

Unit , Integration and Acceptance Test.

PACT – Tool that is used as a way to test contract from microservices that are connected to external APIs ( It is an opensource tool ) .

OAuth – Open Authorization 🡪 to authorize between 2 systems. Facebook services in another site.

Docker is used for deployment . You can add all the dependencies you need for your microservice in a docker image and then we use that for deploying to RAP .

Semantic Monitoring : Testing the different business cases . Eg: e-commerce site , you are testing checkout and such business use cases.

CDC ? -> Client Driven Contract . Aware of who the client ? what changes they need ? and what changes you need to make. Focusing on the client.

Docker – Containerization , image to deploy on fresh VM . Scale up your deployments by wat the mcsebcs needs software and all. Packaging the micrservces and deploy the docker image.

Reactive Extensions🡪 To write async code using Observable design pattern/ Streams .

We can do Continuous Monitoring 🡪 keep checking the health of system. Implementing health check , memory utilization .

Create duplicate instances so that app never goes down. Load Balancing , it uses service delivery / ribbon [.@EnableEurekaClient](mailto:.@EnableEurekaClient) and@LoadBalanced , these will create multiple instances and then return the running one if one is down.

@**LoadBalanced** 🡪 Tells the Euerka server to find the service from the name that we pass ( so instead of hardcoding the url you can mentioned the name ) .--> In a way it does service discovery.

Meaning if you have started the service on 2 ports ( 8080 & 8081) then you see 2 instances in eureka server. So the client has to select one that is what @LB does.

And Eureka client does by default is pings in intervals called heartbeats stating that I am alive. And if no response then makes it DOWN.

And if the discovery service is down it picks it up from cache. 🡪 all this is done by eureka loadbalancing and rest template. And is FAULT TAULARENT .

If MS is slow🡪 Due to threads in webserver . It assigns a thread of each request that comes in . Soon it get exhausted, so we can solve this problem is adding timeouts. One way 🡪 As we use SpringRestTemplate(using HTTPcomponentClientHttpRequestFactory.setTimeout(3000) to make API calls, we can give timeouts. If we don’t get response within time ,we can end it there and return an error. Second way 🡪 As if we have requests coming in every 1 sec but timeout is set for 3sec. So best solution would be to NOT CALL the MS . ( Hold the requests ) This is called the Circuit Breaker Problem 😊 .. So it should do something when it cannot do what is asked of . We implement this like this : First we need to know when to break the circuit , logic has to be smart and know the parameters as Look at the last n request for failures, and how many of those should fail eg 5, timeout duration (for the request ) , and sleep for 10 sec i.e. break the circuit. And then throw an error/return a default response /Save the previous response in cache and use when required so if the request that comes in asks for the same data when the MS is dont but because we have in cache we can return the same and it won’t even know the MS is dont, and continue for the consequent requests.

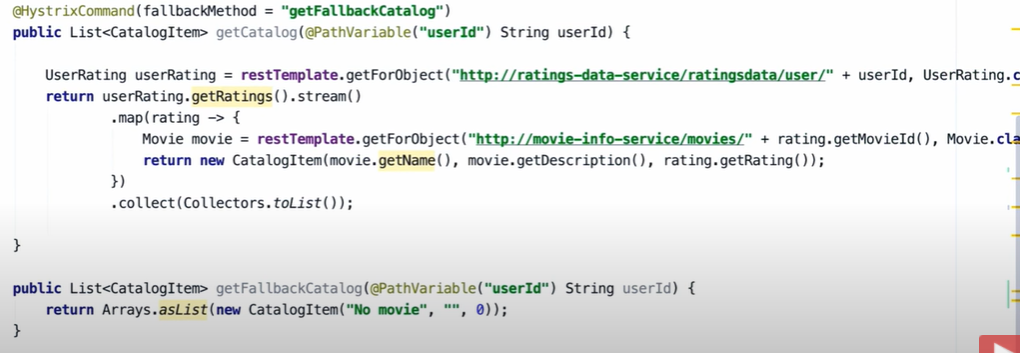
Graphical user interface, text, application

Description automatically generated

Hystrix 🡪 is used to implement the circuit breaker problem so that you don’t have to do it by setting the configuration parameters. Add @EnableCircuitBreaker to the application , @HystrixCommand to the methods that need circuit breakers and lastly configure the behavior.

We can use HystrixDashboard to best understand the MC behavior . Add the dependency and @EnableHystrixDashboard. And add this in system.properties for endpoint for the dashboard :





MS DP : Saga

Spring Security configuration – Spring Security-OAuth , SAML , cloud security, kebernos.

**DelegatingFilterProxy->Servlet->Filter->Authenticaton(crds)->Principal(result)**

We can authenticate – using Authentication Providers->

Diagram

Description automatically generated

Authentication and Authorization :

<https://www.javainuse.com/spring/boot-jwt>

**//while creating the token -**

**//1. Define claims of the token, like Issuer, Expiration, Subject, and the ID**

**//2. Sign the JWT using the HS512 algorithm and secret key.**

**//3. According to JWS Compact Serialization(https://tools.ietf.org/html/draft-ietf-jose-json-web-signature-41#section-3.1)**

**// compaction of the JWT to a URL-safe string**

private String doGenerateToken(Map<String, Object> claims, String subject) {

return **Jwts**.builder().setClaims(claims).setSubject(subject).setIssuedAt(new Date(System.currentTimeMillis()))

.setExpiration(new Date(System.currentTimeMillis() + JWT\_TOKEN\_VALIDITY \* 1000))

.signWith(SignatureAlgorithm.HS512, secret).compact();

}

**//validate token**

public Boolean validateToken(String token, UserDetails userDetails) {

final String username = getUsernameFromToken(token);

return (username.equals(userDetails.getUsername()) && !isTokenExpired(token));

}

You can use annotations to instruct Spring what to do: expose endpoints, wrap methods in transactions, intercept methods in aspects, and so on. Also, you'd like to apply security configurations. This is where Spring Security comes into action.

Spring Security components intercept the requests and make sure that who makes the requests has the permissions to access specific resources. The developer has to configure them in a way that they do precisely what's desired.

Other responsibilities that these components have also relate to the storing of data as well as transiting data between different parts of the systems. By intercepting the calls to these different parts, the components can act on the data. When the data is stored, these may apply encryption or hashing algorithms.

OAuth (Open Authorization) is a simple way to publish and interact with protected data.  
It is an open standard for token-based authentication and authorization on the Internet. It allows an end user's account information to be used by third-party services, such as Facebook, without exposing the user's password.  
The OAuth specification describes five grants for acquiring an access token:

* Authorization code grant
* Implicit grant
* Resource owner credentials grant
* Client credentials grant
* Refresh token gran

### CSRF : Configure CSRF Protection(Cross-Site Request Forgery (CSRF)

The next step is to include Spring Security’s CSRF protection within your application. Some frameworks handle invalid CSRF tokens by invaliding the user’s session, but this causes [its own problems](https://docs.spring.io/spring-security/site/docs/5.0.x/reference/html/csrf.html#csrf-logout). Instead by default Spring Security’s CSRF protection will produce an HTTP 403 access denied. This can be customized by configuring the [AccessDeniedHandler](https://docs.spring.io/spring-security/site/docs/5.0.x/reference/html/core-web-filters.html" \l "access-denied-handler" \o "15.2.2 AccessDeniedHandler) to process InvalidCsrfTokenException differently.

As of Spring Security 4.0, CSRF protection is enabled by default with XML configuration. If you would like to disable CSRF protection, the corresponding XML configuration can be seen below.

<http>

*<!-- ... -->*

<csrf disabled="true"/>

</http>

CSRF protection is enabled by default with Java Configuration. If you would like to disable CSRF, the corresponding Java configuration can be seen below. Refer to the Javadoc of csrf() for additional customizations in how CSRF protection is configured.

*@EnableWebSecurity*

**public** **class** WebSecurityConfig **extends**

WebSecurityConfigurerAdapter {

*@Override*

**protected** **void** configure(HttpSecurity http) **throws** Exception {

http

.csrf().disable();

}

}

Headers in Spring Security:

Caching

Content type

http strict transport security

Remember Me functionality -

rememberMe().key("uniqueAndSecret").tokenValiditySeconds(86400)

The **Remember Me cookie** contains the following data:

* ***username*** – to identify the logged-in principal
* ***expirationTime*** – to expire the cookie; default is 2 weeks
* **MD5 hash** – of the previous 2 values – *username* and *expirationTime*, plus the *password* and the predefined *key*

The first thing to notice here is that both the *username* and the *password* are part of the cookie – this means that, if either is changed, the cookie is no longer valid. Also, the *username* can be read from the cookie.

**To require HTTPS for the login page** modify your security configuration by adding the following:

http.requiresChannel()

.antMatchers("/login\*").requiresSecure();

Or add the *requires-channel=”https”* attribute to your XML config:

<**intercept-url** pattern="/login\*" access="permitAll" requires-channel="https"/>

**the entry point to check if a user is authenticated and logs the person in or throws exception (unauthorized)**.

#### **SocialApplication.configure(...)**

@Override

**protected** **void** configure(HttpSecurity http) **throws** Exception {

*// @formatter:off*

http.antMatcher("/\*\*").authorizeRequests().antMatchers("/", "/login\*\*", "/webjars/\*\*").permitAll().anyRequest()

.authenticated().and().exceptionHandling()

.**[authenticationEntryPoint](https://www.tabnine.com/code/java/methods/org.springframework.security.config.annotation.web.configurers.ExceptionHandlingConfigurer/authenticationEntryPoint)**(**new** LoginUrlAuthenticationEntryPoint("/")).and().logout()

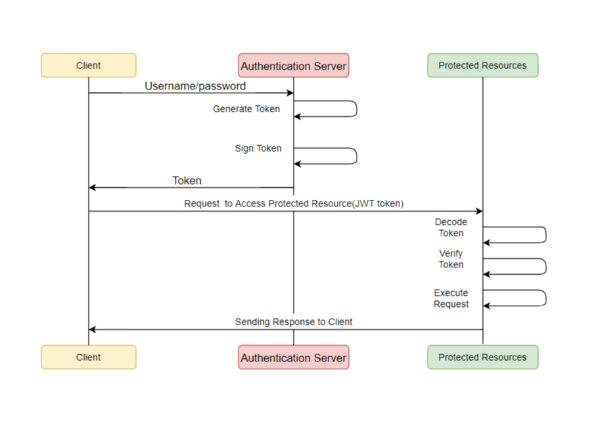
.logoutSuccessUrl("/").permitAll().and().csrf()

.csrfTokenRepository(CookieCsrfTokenRepository.withHttpOnlyFalse()).and()

.addFilterBefore(ssoFilter(), BasicAuthenticationFilter.**class**);

*// @formatter:on*

}



**Spring Filters and Interceptions:**

https://eclipse-ee4j.github.io/jersey.github.io/documentation/latest/filters-and-

interceptors.html

Diagram

Description automatically generated

HandlerIntercepors, on the other hand, intercepts requests between the DispatcherServlet and our Controllers. This is done within the Spring MVC framework, providing access to the Handler and ModelAndView objects. This reduces duplication and allows for more fine-grained functionality such as:

* Handling cross-cutting concerns such as application logging
* Detailed authorization checks
* Manipulating the Spring context or model

CICD - 1.Design – define the stages –

2.which build tool to use ?

3.deploy? (aws/azure/gcp)

Pipeline 🡪 runner () -> runs the stages individually[if for build stage it needs 3 folders and compile and all what ever is needed for the stage to completed successfully ] and sequentially .

Runner will delete resources created in the previous stage(eg: build i.e. folders) before moving on to the next stage.

Solid Principal – Implementations and how and why we use .

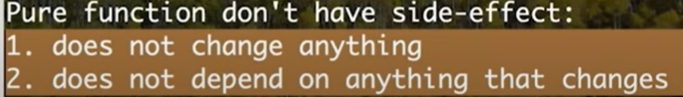
Functional programming – Assignment less programming. OOP doesn’t deal with mutable state .

Goto is to structured programming | assignments is to functional programming.

We can do object composition and decomposition similarly we can do function comp and de comp . those are higher order functions.

Text

Description automatically generated

Functions are pure -> Don’t have side effects i.e. 

Benefits : it reduces accidental complexity . it reduced code.

Java is Functional Style language as it provides higher order functions and also works with mutability.

How we use Lambdas as functional style programming :

Text

Description automatically generated Graphical user interface, text

Description automatically generated

So thread is a higher order function as it takes a function,in this case is a lambda function.

We can use type inference i.e. static meaning it is checking at compile time. And so we can remove the type dependance.

Text

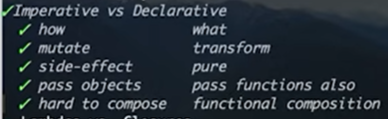
Description automatically generated

Declarative and Imperative style of coding : For eg ,here we see mutability is more in Imperative .

A picture containing text, grass, mammal

Description automatically generated

FP : We are not mutating any variable . It removes the accidental complexity . Here is focus on what to do and in IP: How to do things in details .



When we use variables in Lambas they are effectively final :

Text

Description automatically generated

But you can break this :

Text

Description automatically generated

Text

Description automatically generated

Benefits :

Code Clarity

Fewer Errors 🡪 Doesn’t have accidental complexity , less mutability , removs the extra garbage variables .

Easier to parallelize. 🡪 code structure is same between sequential and parallel , youcan call parallel/ parallelStream()

Copyonwrite/concurrenthashmap --

AWS Tech Stack

API Performance response.

Multithreading

Memory Management

TDD/BDD Approach

GOF Patterns – Intermediate

Singleton- Classic example Runtime ( lang ) which define the runtime of java application.

Private constructor , static object member variable, method to return 1 or null instances.

Break : Cloneable , Serializable , Reflections API

Not to break : Override clone 🡪 throw exception, override readResolve() method , in constructor throw Exception if instance!=null .

Traditional SDP : Use Synchronized to avoid RACE condition so that threads can access 1 by 1. But if two access at the same time then condition is true for instance is null and so returns true. That’s why we do double checked . i.e. we add synchronized block on instance variable and check for instance to be null to return new .

Race condition occurs when multiple concurrently executing process access a shared data item and result of execution depends on the order in which execution takes place . hence data item may lose consistency. A deadlock is when two (or more) threads are blocking each other.

Graphical user interface, text, application, email

Description automatically generated

One of the best ways to create is using innerclass :

We create a inner class and in that create a private static final instance variable.

And in the base class return that instance.

Graphical user interface, text, application, email

Description automatically generated

Here the get instance is called when invoked the first time and is thread safe I don’t have to explicitly mention synchronized keyword.

Factory DP : To use another class to create objects based on condition rather than depending on the client to call the subclass directly.

Template DP : Create abstract class with concrete and abstract methods . Then implementing those methods where this class is extended. And creating object of sub and calling the method.

Parent child1=new Child1();

Parent child2=new Child2();

Façade DP : It hides the mechanics and shows the main functionality . Eg : Compiler it has all stream / scanner / Tokenizer .

Adapter/Bridge DP : Creating a API i.e. Interface . So we have abstract interface for DB drivers. We can use it to serve multiple purposes.

Strategy DP : Behavioral DP 🡪 Selecting algo at Runtime. Compatible with Open extension /Closed for modification. We are dividing the responsibilities in different classes and define the common methods in Interface. So have behavior in separate classes. Create a abstract class and leave the implementations to the sub classes so that you can extent the functionality without breaking it. Comparator class .

Observer DP : Is where the observers are observing a channel i.e. List<Observers> and the Subject notifies them when any state change . And will be notified when an event occurs.

**Non Tech in Grow :**

Experience in one of the approach TDD or BDD.

TDD is Test Driven Development. This means writing a test that fails because the specified functionality doesn't exist, then writing the simplest code that can make the test pass, then refactoring to remove duplication, etc. You repeat this Red-Green-Refactor loop over and over until you have a complete feature. ( red-green-refactor)

BDD is Behavior Driven Development.( Discovery- what it could do , formulation – what it should do , automation- what it actually does 🡪 repeat to give feedback ) This means creating an executable specification that fails because the feature doesn't exist, then writing the simplest code that can make the spec pass. You repeat this until a release candidate is ready to ship.

TDD is a development practice while BDD is a team methodology. In TDD, the developers write the tests while in BDD the automated specifications are created by users or testers (with developers wiring them to the code under test.)

Delegation – Intermediate

Factors to consider : Project timelines & Project expectations.

Whom to delegate : Skills / Clearly articulate the desired outcome / Match the amount of responsibility with the amount of authority .

Software development process – Advanced

Plan analyze design implement test integrate and maintenance

Scrum/Agile – Advanced

Agile – Terminologies ( Techniques – estimation Techniques / Sprint planning , grooming, backlog, velocity , impediments , how to identify )

Sprint velocity : By looking at the amount of work your team completed in previous sprints, you should be able to estimate how much work they can do in future sprints. In Agile development, this estimate is known as sprint velocity. Like calculate the story points in last 3 sprints , take the average. And prepare velocity charts.

We can improve: Develop a better definition of “done.” , Proper communication, retrospective meetings.

* Estimation Techniques : 1,2,3,5,8,13 (Story points (SP) reflect the effort associated with the user story from the dev teams perspective. SP is used to assess the difficulty/complexity of the story, not the promise of how long it will take.
* Ideal days are the effort days to complete a user story )

Table

Description automatically generated

Code Review Process implementation – We can use gerrit , pre-commit peer review system based on git dev by google.

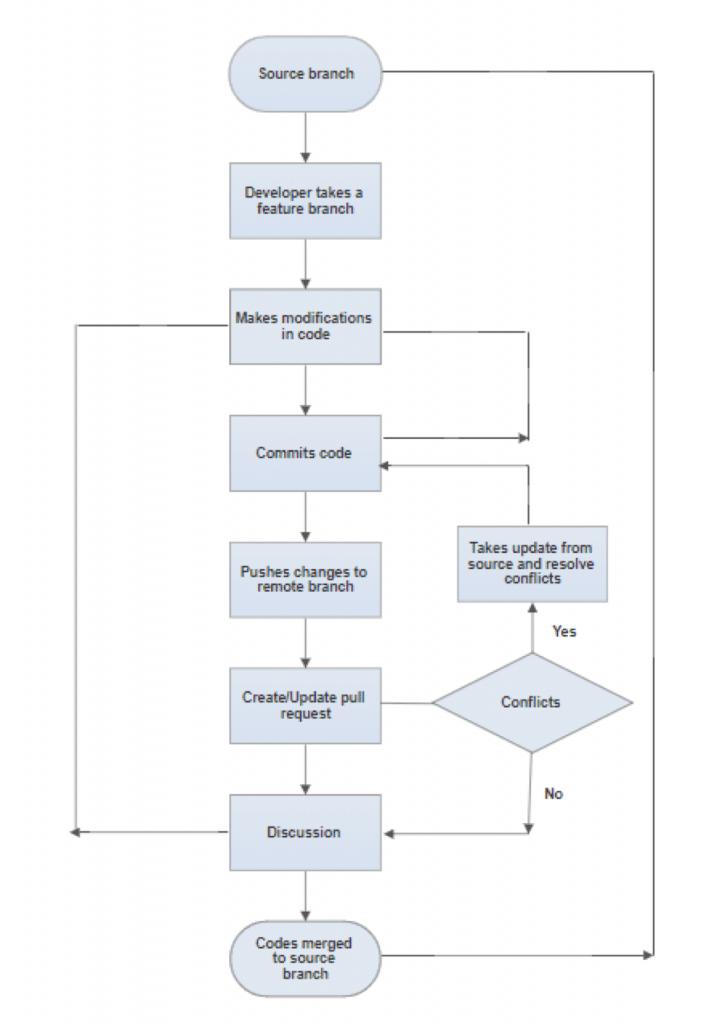
Goals of gerit : Easy to make branches , check if the commit is good ( no compilation errors/no merge conflicts / unit tests passed / code style is fine/no violations ins SONAR/ test coverage) only then it will be merged.

Graphical user interface, text, application, chat or text message

Description automatically generated

Before : Follow the standards , Document it , Rebase your feature branch then push to remote then PR to Source.

After : Discussions, merge conflicts / commit and refactor cycle. And merged to the source branch.



1). How to Improve .Net core application?

2). Junior developing asking controller to view in .Net core?

view bag / view helper

action to action

session mechanism

3). Angular SPA

pros and cons with SPA

4). why you support Angular

5). change detection strategies in Angular?

onpush /change detection

6). difference b/w on push/change detection

7). why change detection

8). generation in .net (garbage collector)

9).why we need generations

when two generations will be checked?

mark & sweep alogorithm

10).any time out in generations? 20 sec time out

11). how to implement caching in .net

local/sql server cache/redis cache

Andrey - technical ,

-------------------------------------

sergrey - desgin and architecture skills

1). describe architecture style applied in current project? – Event driven architecture using Kafka -> for rule engine evaluation )

2). Is it monolith/micro service?

3). why micro service / why not monolithic (*too large* and *unable to be changed* – SaaS would be best for Monolithic *)*

loosely coupling.

4). multi tenancy/multi instance

different instance

multi tenancy ->Software architecture in Cloud Computing | Eg : SaaS ( has multiple users accessing the solution software’s for their purposed across geographical locations | Providing shared hosting | Multiple users access the software at the same time on the same computer | Multitenant applications typically include a level of customization for tenants, such as customizing the look and feel of the application or allowing the tenant to decide on specific access control permissions and restrictions for users.

With VMs, a [hypervisor](https://www.redhat.com/en/topics/virtualization/what-is-a-hypervisor) spins up guest machines that each have their own operating system as well as applications and dependencies. The hypervisor also makes sure users are isolated from each other.

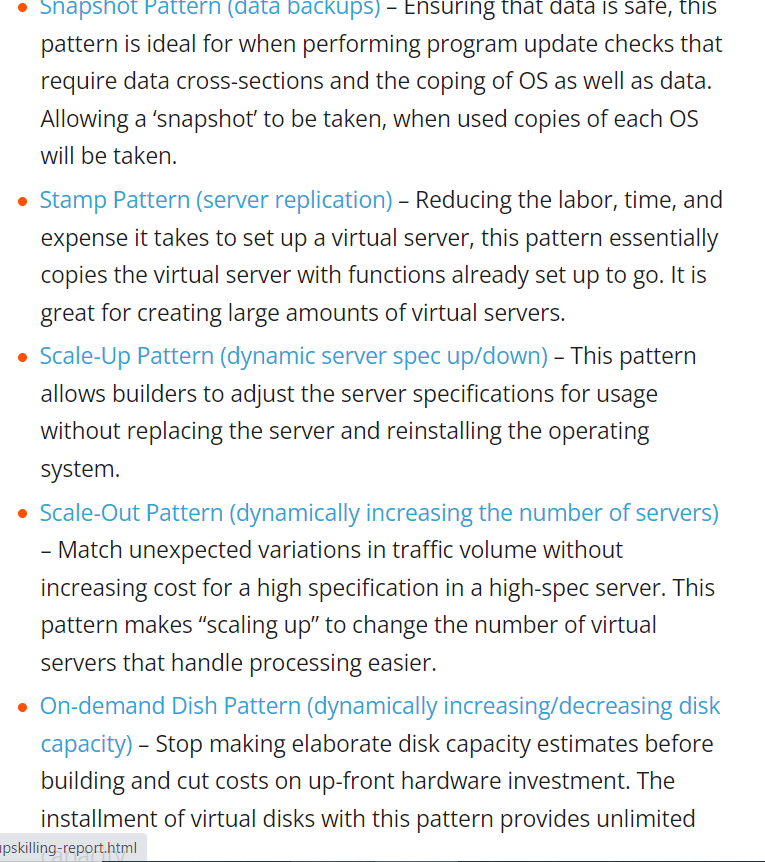
[Compared to VMs](https://www.redhat.com/en/topics/containers/containers-vs-vms), containers offer a more lightweight, flexible, and easier-to-scale model. Containers simplify multi-tenancy deployments by deploying multiple applications on a single host, using the kernel and the container runtime to spin up each container. In contrast to VMs, which each include its own [kernel](https://www.redhat.com/en/topics/linux/what-is-the-linux-kernel), applications running in containers share a kernel, even across multiple tenants.

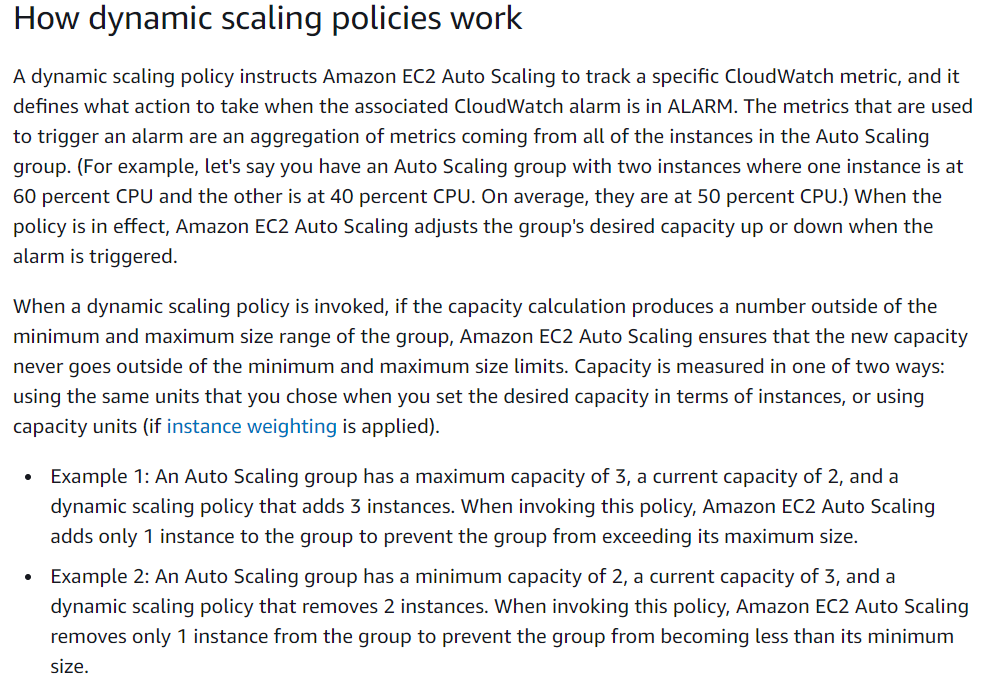
5). patterns of cloud architecture- cloud design patterns .

single table pattern

DP: Behavioral – How the interaction between objects ( Template ) | Structural – What sort of structure you want to follow uses Inheritance | Creational – Advantage of creation of new objects as a solution for a problem and used Polymorphism .

AWS DP :





EC2 - is simply a remote (virtual) machine. ECS stands for Elastic Container Service - as per basic definition of computer cluster, ECS is basically a logical grouping of EC2 machines/instances

Graphical user interface, application, table, Word

Description automatically generated

6). non functional requirement on project

logging

exception handling

where non functional requirement coming from? -- Concentrates on the user’s expectation | Helps you to verify the performance of the software | Describes how the product works

what is the weightage of requirement of non Functional requirement.

7). slide 8

why msmq --

8). recent architecture pattern you participated – Microservices / MVC

about single table approach

how did you do research - Udemy

9). you have design web application , 99% need to be available , which design pattern to use? – I would use Loadbalancing using Zuul (API Gateway) ad Eureka (Discovery Server - Meta space about the services) for 0 downtime of the applications.

<https://www.credera.com/insights/zero-downtime-rolling-deployments-netflixs-eureka-zuul>

<https://medium.com/@dantwining_26268/zero-downtime-blue-green-deployments-for-microservices-7896558623b2>

Cloud provide SLA's – Service Level Agreement -> how much time to generate response : 200ms. ( SLA acts as a future foundation for provisioning and monitoring of services in cloud computing. Consumers need SLAs to specify their requirements regarding quality of service, security, and a backup plan for performance failure)

I would use MVC Design Pattern.

10). regarding documentation /flow on page 9

why/whom this was created

we have only 10 minutes?

11).situation one - improve the documentation state, what steps we considered? – Reference to the old documents.

-------------------------------------

kyrylo korobov - roles create on project sys one

* 1). senior engineer & lead software title what are the responsibility - Your goal will be to develop high-quality software that is aligned with customer needs and business goals. Ensure maintenance of the organization’s operating systems and other technical equipment in use.
* Review the performance of the operating system used in the Company and upgrade it as and when required. Plan the implementation of a new OS when needed
* Document and demonstrate solutions by developing documentation, flowcharts, layouts, diagrams, charts, code comments and clear code
* Remain current on new technologies and available vendor packages, evaluate and make recommendations as necessary
* Assist in task planning, estimation, scheduling and staffing
* Mentor junior and mid-level engineers
* Grow engineering teams by interviewing, recruiting and hiring
* Stay on the leading edge of development practices
* Work in close partnership with cross-functional teams and management

2). you are suppose to become a lead s/w engineer?

biggest achievement that you done as a lead s/w engineer than s/w engineer

Bouncy Castle for Security tool for FPT Users & History Replay 3 fold revenue generation enhancement for the cust

a). take responsibility

b). take along with me

c). suggest sonar /static code analysis / delivery time and better experience with new project than the legacy systm

d). architecture decisions

3). consider your self as team lead / tech lead guy?

4). where you want to move as a delivery manager/solution architect?

5). what is important in mentoring/ what should you do/ what great qualities that you need? – consistency/ patience/ effective communication ( on the same boat )

6). Grow form -> transfer phase -> document model

unit testing/automation testing

style guidance

-------------------------------------

andrey - quality part

1). how to ensure code quality in the project.

sonar bug / Fortify / JMeter / Checkstyle / gerrit .

cyclomatic complexity – Time and space complexity

unit test/code coverage -

more coverage/more quality? no more scenarios

2).quality metrics (do you use sonar) 🡪 Sonar Lint ( detect redundancy and duplication of code , reduce application size, code complexity, time and cost of maintenance, and make code easier to read and understand ) and Sonar qube

3). what about security vulnerabilites?

add cors ( Cross origin resource sharing )

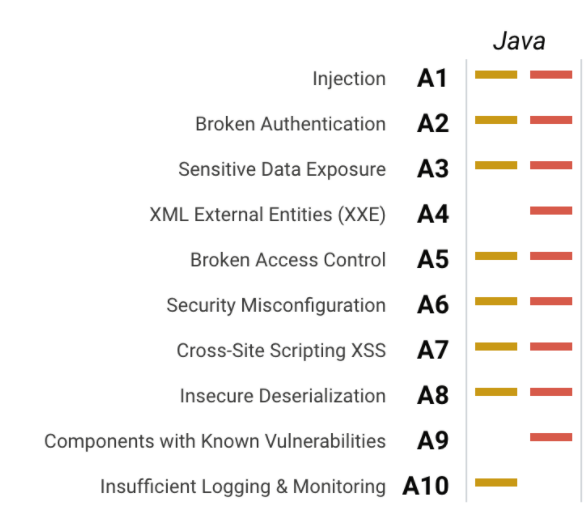
csrf – cross script request forgery

anti forge

JWT token

SSL certificate

directory vulnerabilities : json / yml



integrate pipe line (similar to sonar cube) for security

4). what scenario do you consider unit testing?

fast

AAA - Arrange, Act, Assert.

FIRST- Fast Independent Repeatable Self Validating Timely

Simple and easy to test with border case conditions

act assert

5). how as a team leader make sure is following quality policies in project

run static code analysis

dev pipeline -> sonar cube integrated

write technical debut -> improve development

Focus on requirements

Follow project processes

Document lessons learned

Take time for a through project de-brief

6). TDD -

you have project and test cases are less (lots of legacy)

they don't have (or) too experienced and number is increasing

7).test coverage % -> good or enough

is it same for every project- Code / branch/condition.

8). Engx in EPAM( <https://www.epam.com/engx360> )

how it is being used in EPAM – KPIs / Delivery – CI / Agile frameworks .

git pipeline setup / git lab .

coding kata events

-------------------------------------

sergrey - software process knowledge

x

1). how do you ensure /comply with dev standards or procedures?

code review standards

scrum events

estimations

2). one developer refuse to use coding standards?

3). describe s/w development process

<https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban>

agile / how you follow agile

4). different approach that you worked with apart from scrum

kanbann approach -> how do you make sure urgent thing

do you collect any metrics

velocity -> burn down chart

<https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban>

5). have you done any suggestions/improvement to the process.

6). sitation in your team

responsible for 3 features takes 1 month

customer is asking when 1 of them going to available in production

7). estimation techniques

using fibono series/planning poker

if you want to give estimate large chunk

man hours/story points -> why?

-------------------------------------

sergey - situation questions

1). there is a sprint back log, how you proritize all tasks are important

2). legacy application and team is frustrated

how you can improve

they say i can not advance -> tell importance of grow / give learning parth/ how they important

3). you see a risk, what you do with risk

how to mitigate the risk - Discuss with peers – Roadmap for mitigating the issue.

4). based on what factors you delegate the task

senarioty index

-------------------------------------

kyryko - customer relations

a). did you sprint review with customer relations

b). what we need demo , what is important

c). situation, following scrum approach having demo session with po

it is appeared as demo works as good but not met with customer feedback(-ve feed back)

negative feedback

how to improve -> requirement gap/

story what you think

requirement flaw

d). some reason you can not deliver

due to sick leave/risk/

customer asking over time -> what you do

e). which situation you will escalate to project manager/delivery manager

1.Timely deliverables