**Day 1 – Thurday July 21, 2022**

CI/ CD at Liberty Mutual

Continuous integration and continuous development

Unit tests- piece meal testing. A small unit of whatever I am building, test for bugs.

Integration tests-

Jest is a JavaScript testing framework built on top of Jasmine and maintained by Meta.

CI/CD keeps the app in a deployable state.

A DevOps artifact is **a by-product produced during the software development process**. It may consist of the project source code, dependencies, binaries or resources, and could be represented in different layout depending on the technology.

Working definition:

1: When the code is changed and you attempt to merge it, it triggers a state of automated tests. Metadata is generated which can be tracked so we can see all the data we need about a particular build. This enables teams to be working continuously and avoid merge-hell. We can always have something ready for deployment and have something ready to show.

2: Allows team to work continuously and catch bugs more frequently, as they come up, as opposed to at the end. When we commit code, bamboo creates tests to ensure our code is operational. If code fails, we can role back to previous versions and find errors.

3: Analogy to playing a videogame, If your character dies, do you want to restart from the beginning after playing for 3 hours, or would you rather pick up from you last save?

Agile

SDLC – Software Development Life Cycle

1. Planning
2. 2. Requirement Analysis
3. Design
4. Implementation coding
5. Testing
6. Deployment
7. Maintenance

Waterfall- long feedback loop. We cannot stop until we deliver the full project. More rigid and stricter process structure

Agile- start with MVP (minimum viable product) and continuously work w/ customer. It is more flexible and continuously evolving

Twelve Principles

The entire list of twelve principles proposed in the Agile Manifesto is:

1. Customer satisfaction through early and continuous software delivery
2. Accommodate changing requirements throughout the development process
3. Frequent delivery of working software
4. Collaboration between the business stakeholders and developers throughout the project
5. Support, trust, and motivate the people involved
6. Enable face-to-face interactions
7. Working software is the primary measure of progress
8. Agile processes to support a consistent development pace
9. Attention to technical detail and design enhances agility
10. Simplicity
11. Self-organizing teams encourage great architectures, requirements, and designs
12. Regular reflections on how to become more effective ( [Principles](http://agilemanifesto.org/principles.html), 2001)

**Day 2 – 7/22/22**

What is SRE? How would you explain it a 5 year old /grandmother? What does site as an application depend on? What is the key to SRE?

5 year: Make sure roblox is always up

SRE is a set of principles that all employees are expected to adhere by, however some companies have staff specialists for this role.

SRE focuses on building reliable systems while DEvops focuses on infrastructure

SREs are needed to make sure apps are functioning w/ minimal issues

Why is SRE needed? Explain Scalability, Availability, Incident Response, Automation.

As the company grows, we need systems to handle the increasing work effectively. We need the System to work as expected. We need to have a process in place for handling issues that occur within the system.

Automate SDLC workflow for efficiency and budget saving. Use developer’s expertise to focus on other higher priority issues

SRE- reliability and keeping customer’s faith and trust

What’s the Difference Between DevOps vs SRE? Explain to 5 year old/grandma?

What are the Typical activities in SRE?

Maintaining the website, responding to incidents, cross-department collaboration

What is the advantage of SRE?

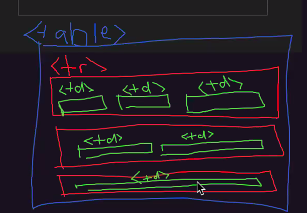
How they mitigate risk and ensure that we don’t lose trust in users and keep business profitable.

**Security Wiley-Edge presentation**

|  |  |
| --- | --- |
| OWASP- Open Web Application Security Project | Regularly compiles common web app vulnerabilities (top 10) |
| CWE- Common Weakness Enumeration | (top 25) Dangerous Software weakness (most common and impactful issues experienced in the last 2 calendar years) |
| MITRE ATT&CK | TTP – tactic techniques and procedures  Globally-accessible knowledge base of adversary tactics and techniques based on real-world observations. Used as a foundation for the development of specific threat models and methodologies in the private sector, in government, and in the cybersecurity product and service community. |
| Application Attacks | Won’t know your areas potential for compromise until you’re compromised.  Indicators of Application Attacks:  -privilege escalation- audit log  -error handling- “undefined exception”  -improper input handling – unsuccessful attempts may be revealed through unexplained crashes |
| Security Coding Techniques | Limit attacks to acceptable risk as outlined by organization  -input validation: causes some of most frequent attacks. If you validate inputs, you can mitigate 50% of incidents. You can do it upfront . Aka (block/allow lists), only allow known-good values.  Sanitation vs. Validation  Sanitation- making sure data is ready for specific use  Validation- data is in the right format. If not valid, then it needs to be sanitized so it is safe for intended purpose. |
| Input Validation- Validating inputs | -check to see if format of data is correct  -syntax errors when data doesn’t conform to expected formal  -semantic- not semantically valid (zipcode is not 5 digits long)  -logical errors- data is semantically valid but doesn’t meet business requirements. (zipcode isn’t valid) |
| Sanitizing Inputs | Input sanitation is the procedure of removing/ replacing invalid characters from data.  Remove invalid characters: approach removes invalid characters from data |
| Normalization | Type of canonicalization that addresses specific issues like encoding, escaping, and data transformation  “1/2/3” needed, “01/02/03” given, normalized to “1/2/3” |
| Output encoding | Makes sure that data is suitable for display on screen to users  Example: URL coding: space character would be replaced with “%20”  Javascript encoding: “the \ character would be inserted before the ` character.  Best Practices:  -conduct all encoding on trusted system  Utilizing a standard, tested routine for each type of outbound encoding  -encode all characters unless they are known to be safe for the intended interpreter |
| Session management | Used to keep track of a user’s session, make sure user is authenticated and authorized. Session usually includes session ID to identify user.  -keep track of secure session IDs. Session identifiers should be long and random.  Sessions should not allow two users to be logged in at the same time  Should be able to cancel own session and invalidate session ID when logging out  Sessions should expire after some time  Persistent logins shouldn’t be used b/c it is a security risk.  Best practices:  -Logout functionality  -set “secure” attribute for cookies transmitted through TLS connection |
| Error handling | Error messages need to be sanitized.  Attackers will confuse systems to get error messages with data that can disclose sensitive information (including system details, session identifiers, account info)  For example, “login unsuccessful vs password is invalid”  -log all authentication attempts, especially failures. (can’t try more than 5 password attempts per hour for example) |
| Memory Management | Process of allocating and deallocating memory so that it is safe and secure.  Sensitive data needs more security, more memory space.  Best practice:  -utilize input and output controls for un-truste4d data  Double check that the buffer is as large as specified  -avoid use of known vulnerable functions (e.g. printf, strcat, strpy, etc)  -truncate all input strings to a reasonable length before passing them |
| Code reuse | You will need to validate pre-developed code that is posted to a library to check for vulnerabilities  Best practices:  Only use trusted dependencies  No not use wildcard imports  Use dependency manager like Maven  -restrict users from generating new code or altering existing code |
| Anti-Tampering | Obfuscation is the process of making code or data difficult to be understood by humans.  Code signing is the process of digitally signing code/data to verify the authenticity  Sandboxing is a security measure used to isolate programs and restrict their access to system resources. |
| Tips for TechStarters | -always maintain confidentiality |

HTML

Tables



**Day 3 – 7/25/22**

CSS- specificity

ID > class > element

**Day 4 – 7/26/22**

Cloud computing

-SAAS

* software as a service- pay per use of application software to users
* on demand service
* don’t need to install software on PC.
* Available for multiple end users
* Ex: gmail, google docs, Microsoft office 365
* Subscription
* Pros; universally accessible,, good for collaborative working
* Cons: internet performance may dictate overall restrictions, compliance

-PAAS

* platform as a service
* web service w/ database
* developers use this
* aka: amazon web services, Heroku
* pro: faster market for developers
* cons: migration issues, developers limited to languages

-IAAS

* System Admins
* Ex: amazon EC2
* Pros: cloud provides infrastructure, flexible, enhanced scalability

CRUD- POST, GET, PUT, DELETE

**Javascript**

**-components- a piece of the UI**

**App > NavBar / Profile/ Trends / Feed > Tweet > Like**

**-**

**class Tweet{**

**State={**

**};**

**Render(){**

**}**

**}**

-React reacts to state changes / updates the DOM