SECURE CODE

WARRIOR

Taking a Preventative Human-Led Approach to Software Security and Embedding It into the Developer’s DNA

Jaap Karan Singh

jaap@scw.io

Co-Founder and Customer Success Guru, Secure Code Warrior

Today’s Agenda

- The ancient history of software creation
- Waterfall vs. Agile vs. DevOps
- DevSecOps: Security as a shared responsibility
- Why security-aware developers are the future
- Transforming from “Dev” to “DevSec”: A tactical guide

Developers and AppSec teams through the ages

The ancient history of software creation

The ancient history of software development.

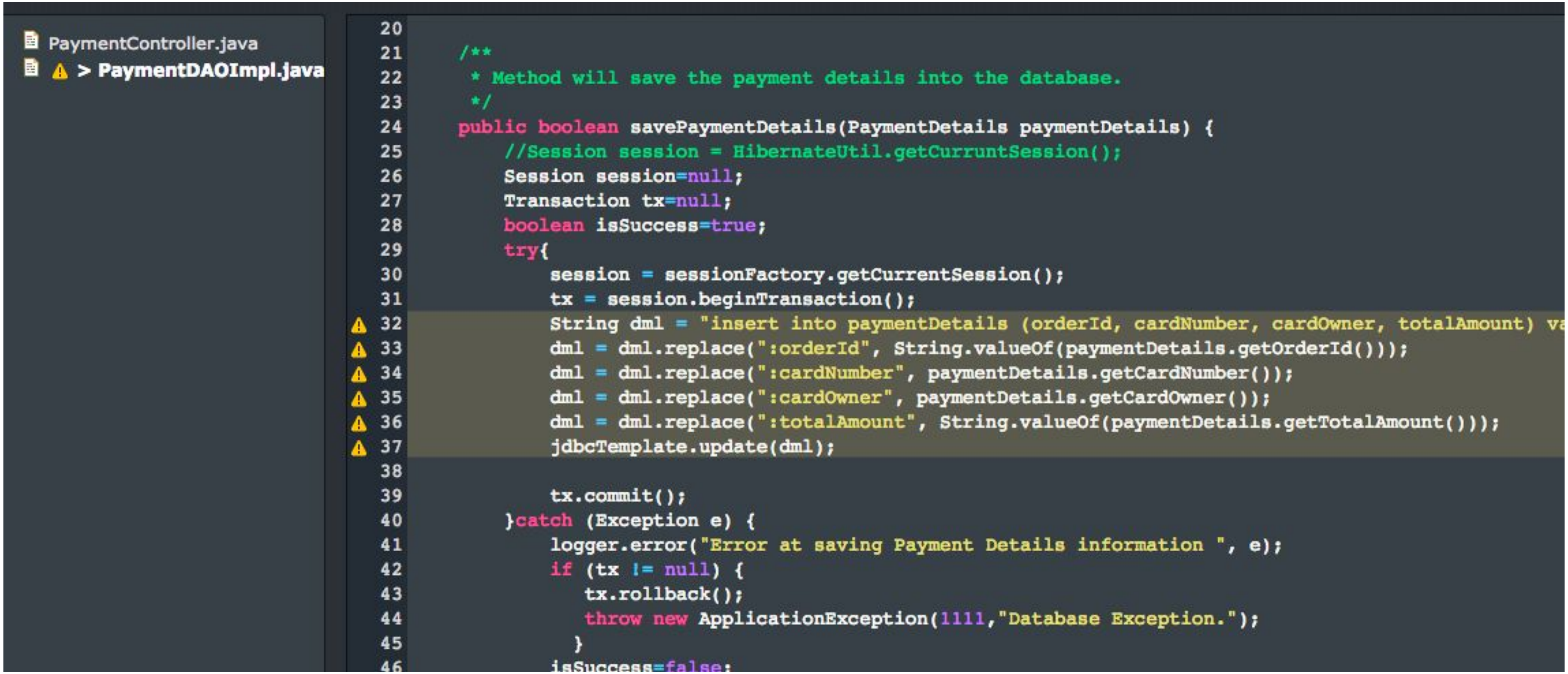
- Developers and Application Security had **little interaction**
- Siloed work environments forged tension
- Developers’ primary goal was creating functional features
- AppSec teams were aware of **security bugs** and issues, but their remedies were often not ideal for the tech stack of the company
- Developers would **delay code review** and shorten feedback windows to limit AppSec interference at last hurdle
- **Bugs discovered late**, or months after public release

The dysfunction continues today.

It’s a feature. Not a bug.
VS
Your code has so many bugs I called Pest Control. Start again!

- AppSec and developers still **get in each other's’ way**
- Most developers **still don’t see security as a priority**
- Security is still being considered **far too late** in the SDLC

Security bugs are everywhere!



```

    /**
     * Method will save the payment details into the database.
     */
    public boolean savePaymentDetails(PaymentDetails paymentDetails) {
        //Session session = HibernateUtil.getCurruntSession();
        Session session=null;
        Transaction tx=null;
        boolean isSuccess=true;
        try{
            session = sessionFactory.getCurrentSession();
            tx = session.beginTransaction();
            String dml = "insert into paymentDetails (orderId, cardNumber, cardOwner, totalAmount) va
            dml = dml.replace(":orderId", String.valueOf(paymentDetails.getOrderId()));
            dml = dml.replace(":cardNumber", paymentDetails.getCardNumber());
            dml = dml.replace(":cardOwner", paymentDetails.getCardOwner());
            dml = dml.replace(":totalAmount", String.valueOf(paymentDetails.getTotalAmount()));
            jdbcTemplate.update(dml);

            tx.commit();
        }catch (Exception e) {
            logger.error("Error at saving Payment Details information ", e);
            if (tx != null) {
               tx.rollback();
               throw new ApplicationException(1111,"Database Exception.");
              }
            isSuccess=false;
```

We can’t afford to continue down this path.

Keeping security bugs under control is harder than ever:

>> 111 billion lines of code is produced each year on average (**and it needs to be secured**)

>> Zero-day cyberattacks are estimated to reach **one per day by 2021**

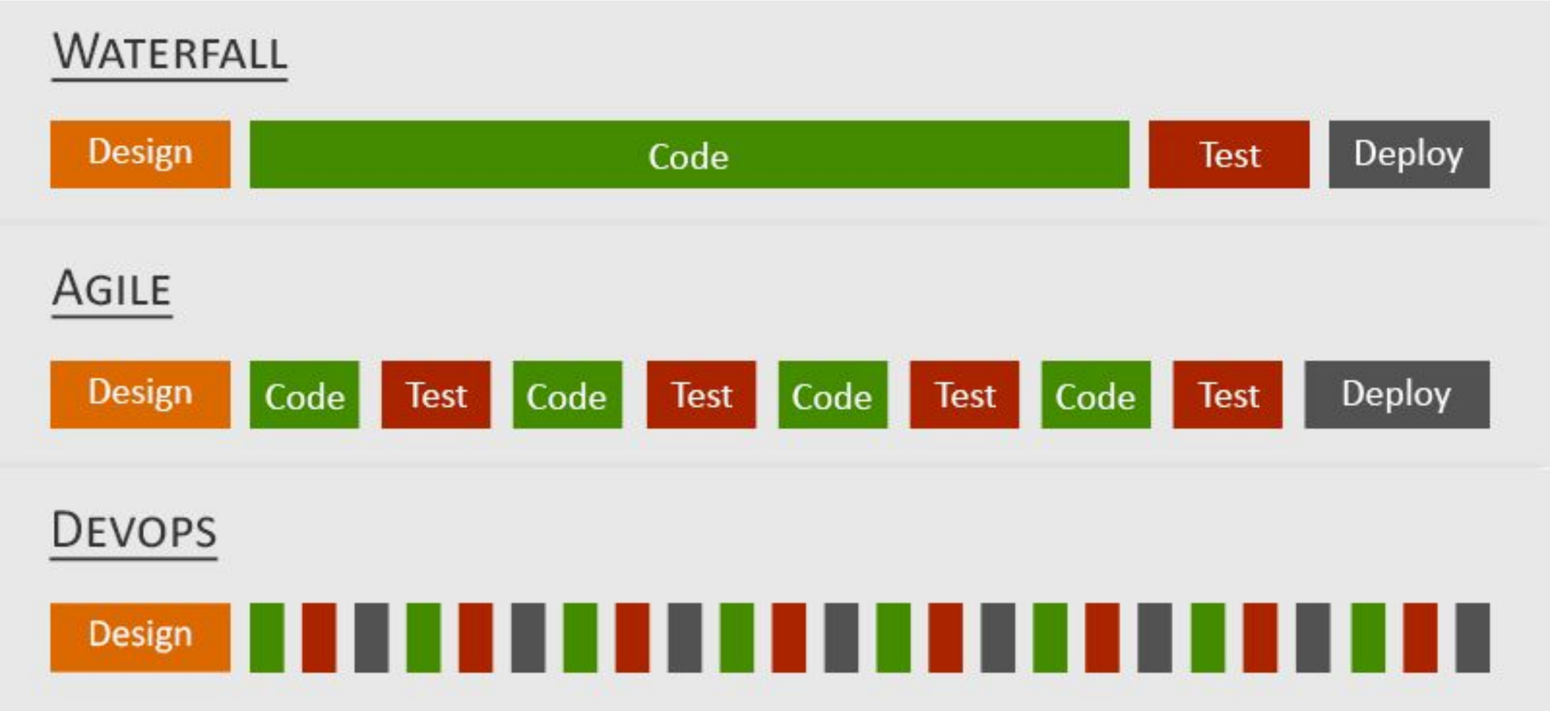
>> The demand for security personnel far outweighs supply (**and will only get worse**)

It’s time to stop the blame game and get to work.

Methodology metamorphosis

Waterfall vs. Agile vs. DevOps

Waterfall vs. Agile vs. DevOps



WATERFALL

Design Code Test Deploy

AGILE

Design Code Test Code Test Code Test Code Test Deploy

DEVOPS

Design

Each stage has improved processes, collaboration and continuous deployment… but security remains back-of-mind.

DevSecOps: The perfect process?

DevSecOps is the latest software development methodology, aimed at keeping security at the forefront of every stage of the project, with *every* team doing their part to ensure software security best practice.

It is:

- An **upgrade** of existing methodologies, taking parts of the Agile and DevOps processes; not a whole new idea
- As much a **cultural shift**, business-wide, as it is a way of producing more secure software
- **Not** a perfect process; as is the case with continuous improvement and deployment in software, so too should our methods of working flex with needs and risks.

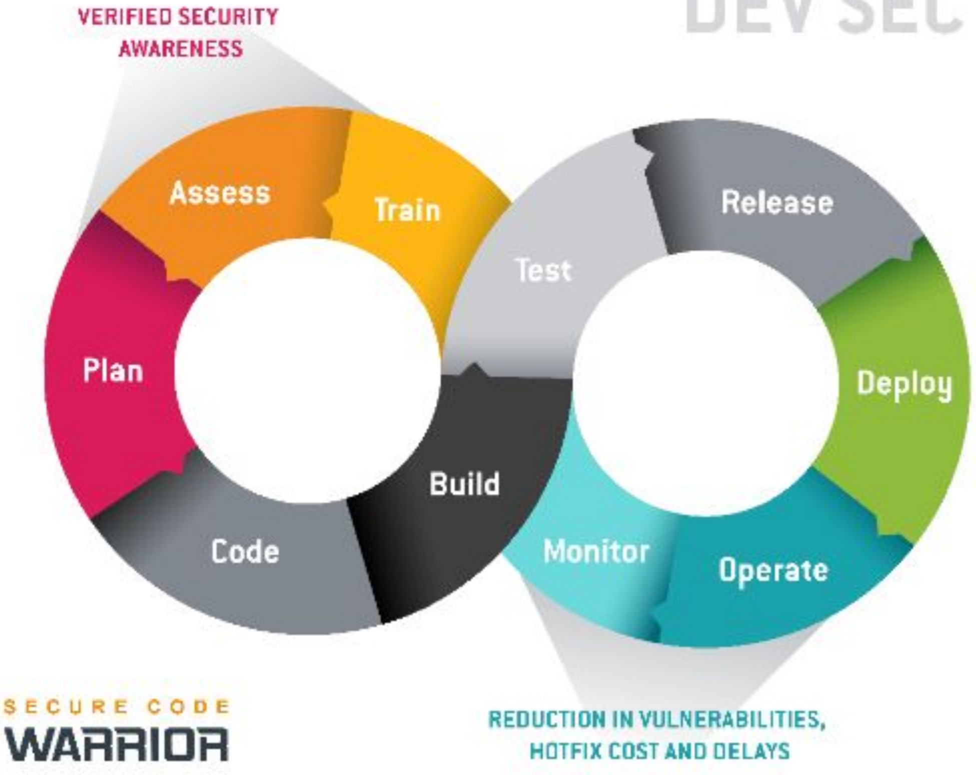
A process that cannot be bought

How is DevSecOps changing the game?

The SSDLC is possible with DevSecOps

THE SECURE SOFTWARE DEVELOPMENT LIFECYCLE (SSDLC)

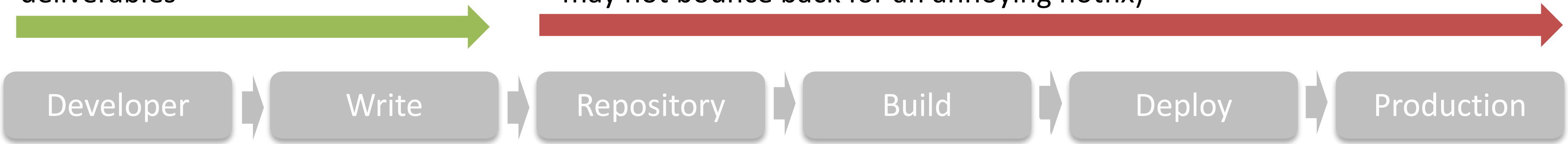
DEV SEC OPS


VERIFIED SECURITY AWARENESS
Assess
Train
Plan
Code
Build
Test
Release
Deploy
Operate
Monitor
REDUCTION IN VULNERABILITIES, HOTFIX COST AND DELAYS

How do development teams do DevSecOps right now?

Developers engage with their deliverables

Magical security fairies scan, find and fix insecure code (and it may or may not bounce back for an annoying hotfix)



Developer → Write → Repository → Build → Deploy → Production

“The code never bothered me anyway...”

Is DevSecOps a game developers want to play?

SECURE CODE WARRIOR

- Developers need to be informed the role they play in new methodologies, and be given the time, support and tools to meet objectives
- DevSecOps doesn’t work if key goals are not agreed upon, or individuals have far too many competing priorities
- Take the time to help development teams early in the process and during the cultural shift.

WORKING TOGETHER
TO BE THE HEROES WE NEED
APPSEC TEAM
DEVELOPERS

The new kind of rockstar

Why security-aware developers are the future

There will never be enough AppSec people to save the day.

Software Developers (Agile)

A
B
A
B
A
B
A
B
A
B
A
B

100

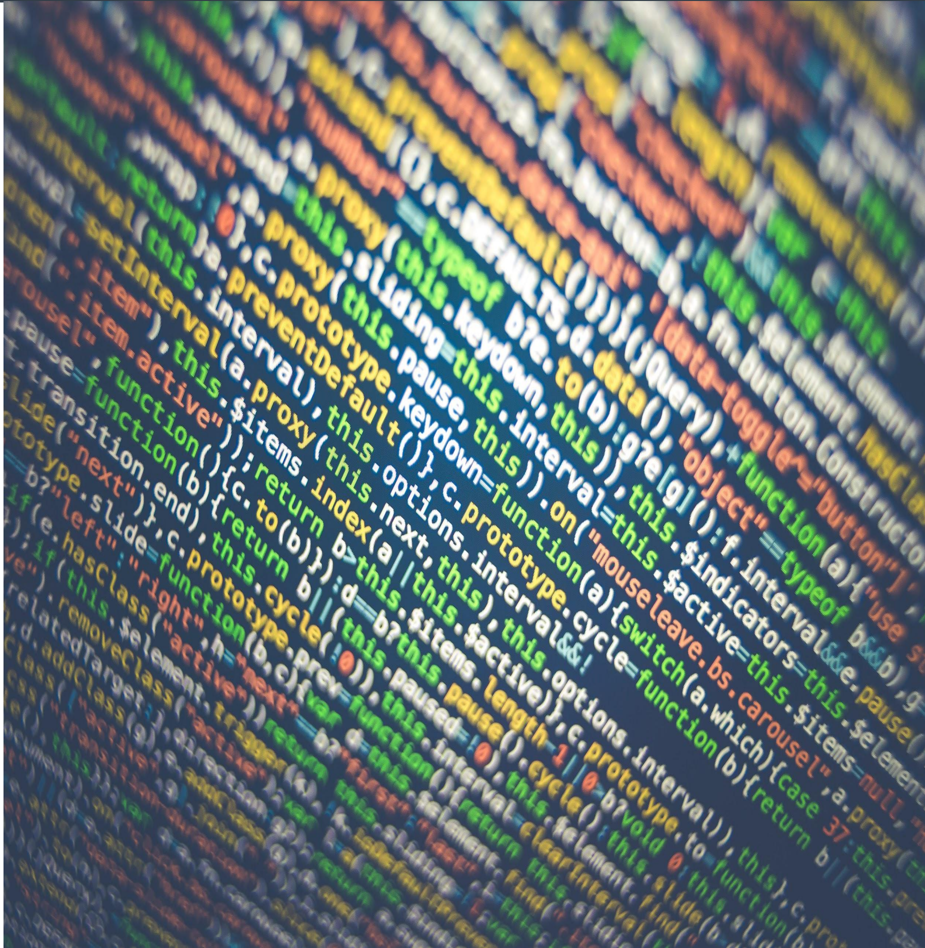
Application Security Experts

1

Securing code as it is written.

Security-aware developers are:

- Highly sought-after (and always will be)
- A cut above average developers
- Presented with more lucrative job opportunities
- Instrumental in the battle against cyberattacks and data breaches
- **Aware that the only good code is secure code**.



Transform with a tactical guide

Making the move from “Dev” to “DevSec”

Security is a shared responsibility

- The DevSec mindset realizes it takes a village to uphold security best practice
- Security-aware developers adjust to quality and safety being just as important as coding speed
- Take ownership of their own coding outcomes, and keep security front-of-mind from the beginning

Tactical Guide Tips

SECURE CODE WARRIOR

Help nurture a highly supportive environment

- ✓ Ensure your KPIs are positioned around secure code, not just speed of production
- ✓ Raise any gaps in knowledge, training or tools that would help you play your part in the DevSecOps process
- ✓ Commit to a higher standard of security yourself, and set an example for others
- ✓ Work positively with other departments: you have the same goals
- ✓ Success depends on support; ensure ALL personnel have adequate time to train and make a positive cultural impact.

Tactical Guide Tips

SECURE CODE WARRIOR

The best DevSecOps teams train constantly, and improve their skills as they work.

- ✓ Developers should be given frequent security training that is relevant to their job and up-to-date
- ✓ Time to train should be allocated from the beginning
- ✓ Challenges must be constantly expanded and updated so developers can continue to build their skills over time
- ✓ Challenges must vary in complexity so they are engaging for both senior developers and less experienced ones
- ✓ Assessments and rewards for upskilling help build a positive security culture and experience

Customer Case Study - Fitting into a developer’s world

Challenge

A Tier-1 US financial institution wanted to upskill their developers on security quickly, scale to hundreds of developers and reduce costs associated with fixing vulnerabilities in the late stages of the SDLC, do all this without impeding developers

Outcome

They required their developers to play a single challenge **(3-5 minutes)** every day for three months. Testing their skills before and after the training period saw a 60% increase in secure development capability across the team. This meant less late-stage bug-fixing and significant long-term savings.

Tactical Guide Tips

SECURE CODE WARRIOR

Successful DevSecOps teams have the right suite of tools for the job at hand.

- ✓ Have the right balance of tools and trained people
- ✓ Deploy SAST/DAST/IAST tools as part of, not the **only** solution, and understand where each is best used in the process
- ✓ Make it easy for company security policies to be accessed, adhered to and updated (e.g. which code libraries to use when designing new software

Tactical Guide Tips

SECURE CODE WARRIOR

The best DevSecOps teams keep it real with each other.

- ✓ The team understand each other’s roles and responsibilities
- ✓ They work with the same toolset to complete the same outcomes
- ✓ AppSec and development teams treat each other as equals
- ✓ The process, training and tools should enhance, not impede, your role as a security-aware developer
- ✓ The team works to understand each other’s challenges and makes compromise where necessary

Summary

Traditionally, the relationship between application security and development has been quite strained. This has led to less than favourable security outcomes.

DevSecOps offers a methodology to make security a shared responsibility through a combination of tooling, cross-team collaboration, and training.

Security aware developers are rockstars and highly sought after because of the value they bring to an organisation.

SECURE CODE

WARRIOR

SECURECODEWARRIOR.COM

INSIGHTS.SECURECODEWARRIOR.COM

@SECCODEWARRIOR

LINKEDIN.COM/COMPANY/SECURE-CODE-WARRIOR

FACEBOOK.COM/SECURECODEWARRIOR/