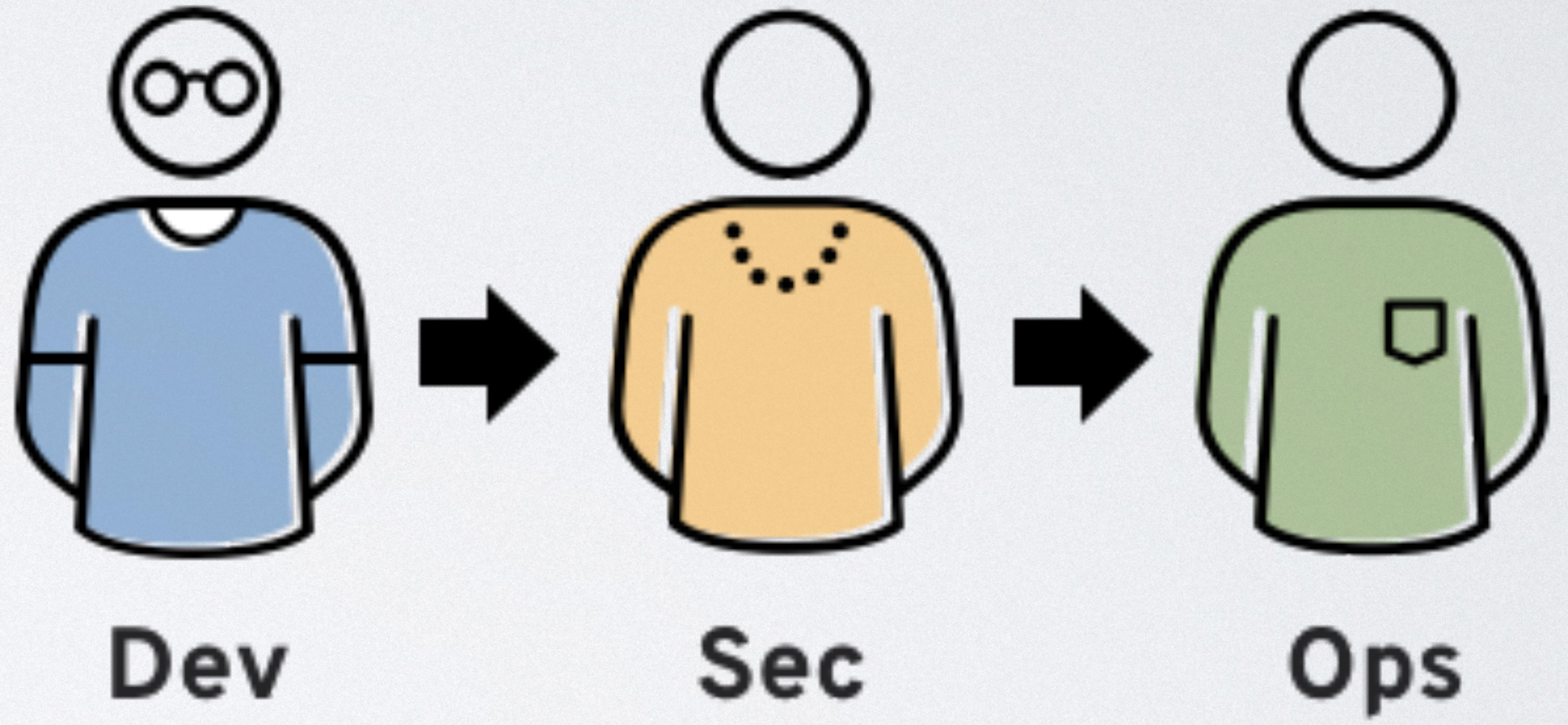
Security and DevSecOps

Integrating Security into the Software Development Process

The Old Way

- First, write the code
- Then, have the security people do their thing
- Then, let the operations people host it
- But doing security too late is bad…



https://www.redhat.com/en/topics/devops/what-is-devsecops

Security Has Architectural Implications

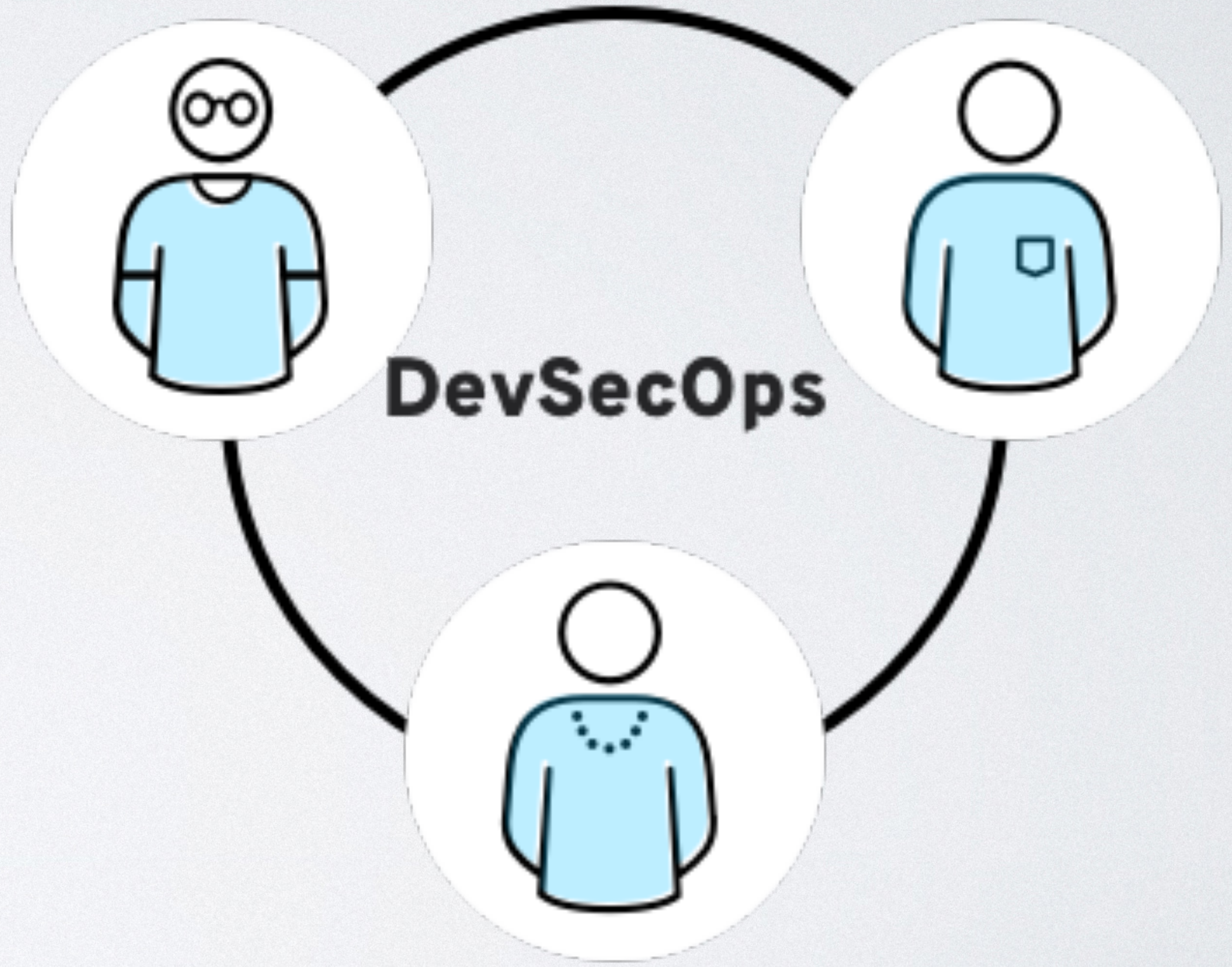
- Where is access control?
- Where is authentication?
- How are credentials passed?
- What are the attack vectors?

More Design Implications

- Tooling: you aren't going to use C/C++, are you?
- Testing processes
  - Penetration tests?
- How will you mitigate social engineering attacks?

DevSecOps

- Integrate security into the development process
- The rest of today: how to include security concerns



https://www.redhat.com/en/topics/devops/what-is-devsecops

Kinds of Security Challenges

Challenge	Approach
Undefined behavior	Don't use unsafe languages (when possible)
Incorrect security-related code	Review, test, control changes
Higher-level design mistakes	Architectural review, penetration testing
Users (e.g., social engineering attacks)	HCI techniques; training; compromise procedures

Microsoft DevSecOps Advice

- Train
- Define security requirements
- Define metrics and compliance reporting
- Use Software Composition Analysis and Governance
- Perform threat modeling
- Use tools and automation
- Keep credentials safe
- Use continuous learning and monitoring

https://www.microsoft.com/en-us/securityengineering/devsecops#Metrics

Train

- Glad you're here.

Define Security Requirements

- Legal and industry requirements
- Internal standards and coding practices
- Review of previous incidents, and known threats.
- Traditional requirements analysis, with security focus

Define Metrics and Compliance Reporting

- How will you know whether you've succeeded?
- Does one breach mean you've failed?
  - Better to focus on progress than success/failure

Threat Modeling

- Goal: enumerate all possible threats
- STRIDE model helps you remember possible threats:
  - **S**poofing identify
  - **T**ampering with data
  - **R**epudiation
  - **I**nformation disclosure
  - **D**enial of service
  - **E**levation of privilege

Exercise

- In groups: enumerate possible threats for your project
  - In a real meeting: spend 2 hours, identify 20-40 issues.

https://learn.microsoft.com/en-us/previous-versions/commerce-server/ee798544(v=cs.20)

Use Software Composition Analysis and Governance

- Vulnerabilities can come via third-party tools and components

Use Tools and Automation

- Tools must be integrated into the CI/CD pipeline.
- Tools must not require security expertise.
- Tools must avoid a high false-positive rate of reporting issues.
- Static analysis
- Dynamic analysis

Keep Credentials Safe

- Scan for keys in source code

Use Continuous Learning and Monitoring

- Continuous integration / continuous delivery
  - Should run analyses automatically
- Mean time to identify (MTTI)
- Mean time to contain (MTTC)

Some Technical Potpourri

Authentication vs. Authorization

- Authentication: are you who you say you are?
- Authorization: Given who you are, what can you do?
  - Policies enforced with access control

https://www.icann.org/en/blogs/details/what-is-authorization-and-access-control-2-12-2015-en

Principle of Least Privilege

- Only authorize access that is actually needed

Origin-Based Restrictions

- Whom does your software trust?
- It might trust other components of your software
  - But not arbitrary third parties