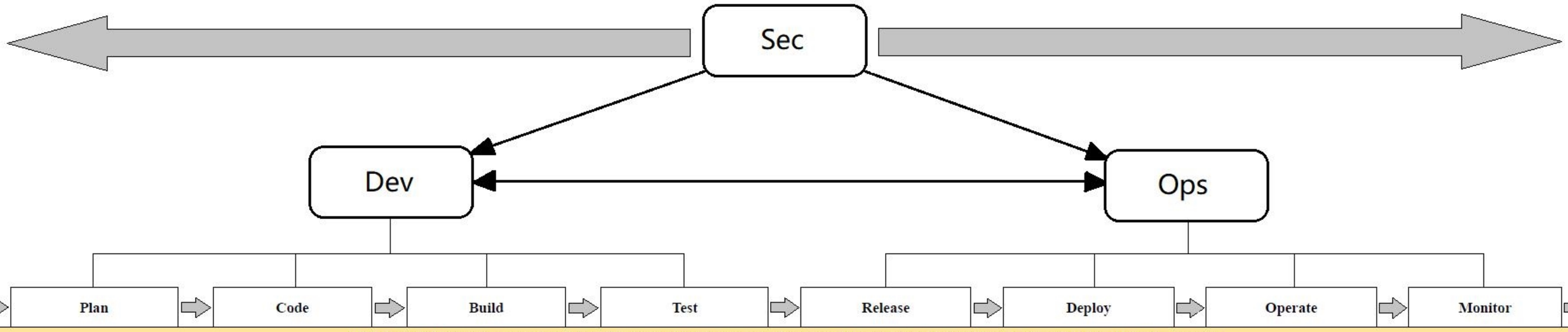

Organization, People & Culture	C02-Challenges of collaboration, communication & coordination—P02-Improving collaboration, communication & cooperation; P08-Continuous feedback loop; P10-Trust within and between the teams; P11-Shameless retrospectives								
	C03-Neglecting security—P06-Security champions; P12-Continuous improvement mindset								
	C08-Lack of security awareness and responsibility—P03-Shared and collective responsibility for security; P15-Pragmatic implementation; P16-Be reactive and responsive								
	C09-Boundary between specialist & generalist—P04-Shared knowledge								
	C01-Cultural resistance, organizational opposition to change—P01-Cultural shift to security								
	C04-Lack of security knowledge and skills, need for training—P05-Training, learning and education for security—M01-Security-trained rate								
	C05-Recruiting challenges—P07-Hiring new personnel and recruiting success								
	C06-Challenges in decision level—P09-Get leadership's supports and buy-in from stakeholders to make security a priority								
	C07-Lacking confidence—P13-Commitment & Agreement								
	C10-Insufficient level of governance on DevSecOps adoption—P14-Enhance transparency								
Process Capabilities	C11-Difficulties in integrating security into DevOps without losing speed—P17-Shifting security to the left—M04-Time spent correcting mistakes in each category; M07-Whether security review slows down the development cycle								
	C12-Difficulties in transforming to DevSecOps without affecting current process—P17-Shifting security to the left—M05-Number of continuous delivery cycles per month; M08-How well security is integrated into the delivery lifecycle								
	C22-Interconnectedness of the DevOps process—P17-Shifting security to the left—M05-Number of continuous delivery cycles per month								
	C15-Lacking security standards—P33-Common weaknesses enumeration (CWE)—M03-Top vulnerability				C13-Compliance requirements—P21-Compliance control; P59-Compliance-as-Code—tools: Nspec, ServerSpec, OpenSCAP	C23-Continuous deployment chaos—P45-Fault injection (chaos engineering)—M18-Penetration test pass rate	C24-Poor visibility of security track record—P19-Increase the visibility; P20-Good documentation, logging and reporting—Logging tools: PaperTrail, Logstash, Loggly, Splunk, SumoLogic		
	C16-Ignoring processes and security essentials leading to technical and security debt—P18-Security by Design				C19-Using unsuitable metrics—P28-Measurement—DevOps performance measuring tool: SAFe DevOps Health Radar		C14-Neglecting change control in security—P22-Risk, vulnerability and incident management—Vulnerability management tools: Defect Dojo, ArcherySec, Rapid7 Nexpose, HackerOne, Claire, Stethoscope, Snyk—M13-Critical risk profiling; M16-Point of risk per device		
	C17-Lacking process and platform for communication, collaboration, and sharing information and feedback				P30-Security reviews—M06-Whether features undergo a security review		C21-Tradeoff between security measures and CI system performance—P34-Operational controls validation and improvement		
	C18-Out-of-sync domains and cross team dependencies causing issues in adopting DevSecOps				P31-Security evaluation—M07-Whether security review slows down the development cycle; How well security is integrated into the delivery lifecycle		C25-Inadequate privileged credentials and access controls causing cyber attacks—P23-Least privilege controls and privileged access management;P24-Continuous certification for DevOps; P35-Keep credentials safe		
	C20-Inconsistent security polices design—P29-Policies; P46-Policy-as-Code				P32-Version control, metadata, and orchestration				
Technology	C27-Challenges of legacy system refactoring—P61-Integrate security issues within your general bug tracker	C26-Lack of mature tools for automation and security—P36-Automation—Automation tools: Ansible, Chef, Jenkins, Puppet, Gauntlt, SaltStack—M19-Whether automated testing covers security requirements.							
	C29-DevSecOps is mistaken for a set of tools—P05-Training, learning and education for security—M20-The use of preapproved libraries, packages, tool chains and processes	C31-Complexity in managing different tools—P36-Automation—Automation tools: Ansible, Chef, Jenkins, Puppet, Gauntlt, SaltStack—M19-Whether automated testing covers security requirements.							
	C30-Lack of tool standards	P55-Secure coding—Automated code review tools: Veracode Greenlight, PMD, DevSkim, FindSecBugs—M10-Code scanning detection rate	P52-Adopting DevSecOps in microservices-based applications						
	C32-Difficulty in simulating production—P38-Threat modeling—tools: IriusRisk, Microsoft threat modeling tool—M14-Number of adversaries per application; M15-Adversary return rate	P37-Security-as-Code P46-Policy-as-Code P49-Immutable-as-Code P50-Design-as-Code P59-Compliance-as-Code		C34-Remaining manual security testing and need for automated testing performance measures—P40-Red team security drills—M17-Number of issues during red teaming drills; P48-Combination of static and dynamic analytical methods; P57-SAST—tools: Kiuwan, Flawfinder, Graudit, Bandit, Spotbugs, SonarQube; P43-DAST—tools: OWASP ZAP, Arachini, BDD Security, Nikto, Radamsa, FuzzDB, Fortify Webinspect; P44-RASP—tools: Fortify Application Defender; P58-IAST—M09-Security test pass rate	P63-Application-level assessment—M14-Number of adversaries per application	C35-Availability and reliability of infrastructure resources, tools, automation, and network bandwidth for fast deployment cycle—P47-Detect existing security flaws; P51-Micro-segmentation; P60-Consumable security services with APIs; P62-Configuration management; P41-Make sure the basics of host and network security are in place—Cyber security tools: Tripwire, Snort—M16-Point of risk per device	C28-Use of cloud and serverless computing brings security complications—P64-CI/CD for patching ; P65-Verify cloud infrastructure—Cloud security tools: Terraform, Snorby threat stack, AppScan on cloud, AWS Security service, ThreatModeler Cloud Edition, Trend Micro Cloud One; P39-Continuous monitoring—Monitoring and alerting tools: New Relic, Nagios Icinga, Graphite, Ganglia, Cacti, Pager Duty, Sensu, Boundry, Pingdom, Suricata; P53-Advanced malware detection—tool: CodeAI; P56-Sensitive information scan—tools: TruffleHog, GitSecrets, Talisman	
	C33-Threat modeling scalability issue—P38-Threat modeling—Threat modeling tools: IriusRisk, Microsoft threat modeling tool	P54-Software composition analysis—tools: Retire.js, OSS Audit, OWASP Dependency Check—M11-Defect density;M12-Defect burn rate	C36-Containers and other tools come with their own risks—P42-Container security—Containerization tools: Docker, Kubernetes; Container security tools: Twistlock, Notary, Aqua Security						
Business	C37-Restructuring due to high cost and considering ROI—P67-Business-driven security—M02-Business metrics (DevSecOps KPIs)		P66-Separation of duties		C39-Difficulty in training users for using advanced tools; C38-Customer readiness for frequent releases—M02-Business metrics (DevSecOps KPIs)	C40-Insufficient resource is not able to cope with abundance of information—P69-Availability and business continuity management—M02-Business metrics (DevSecOps KPIs)			
	C41-Dilemma in selection of business processes for DevSecOps transformation—P67-Business-driven security—M02-Business metrics (DevSecOps KPIs)								
	C42-Conflicting approaches to security and business requirements—P67-Business-driven security—M02-Business metrics (DevSecOps KPIs)								