

2022 ACCELERATE

State of DevOps Report



후원:



목차

01	핵심 요약	03	05	놀라운 사실	55
02	비교 방법	08	06	인구통계 및 기업통계	59
03	개선 방법		07	최종 의견	67
	소개	19	08	감사의 말씀	68
	Cloud	21	09	저자	69
	SRE 및 DevOps	26	10	방법론	73
	기술 DevOps 역량	29	11	추가 자료	76
	문화	37			
04	공급망 보안이 중요한 이유	42			

01

핵심 요약



데릭 드벨리스



클레어 피터스

지난 8년 동안 Google Cloud는 33,000명의 전문가로부터 얻은 의견을 토대로 Accelerate State of DevOps Report를 작성했습니다. 본 연구에서는 역량과 관행이 DevOps의 핵심으로 간주되는 성과를 예측하는 데 어떤 영향을 미치는지를 중점적으로 살펴봅니다.

- 소프트웨어 배포 성과 – 소프트웨어 배포 성과의 4가지 주요 측정항목: 배포 빈도, 변경 리드 타임, 변경 실패율, 서비스 복원 시간
- 운영 성과 – 5번째 주요 측정항목: 신뢰성
- 조직 성과 – 조직이 성과 및 수익성 목표를 충족하는 정도

또한 번아웃과 직원들이 소속 팀을 추천할 가능성과 같은 기타 성과의 기반이 되는 요소들도 조명합니다.



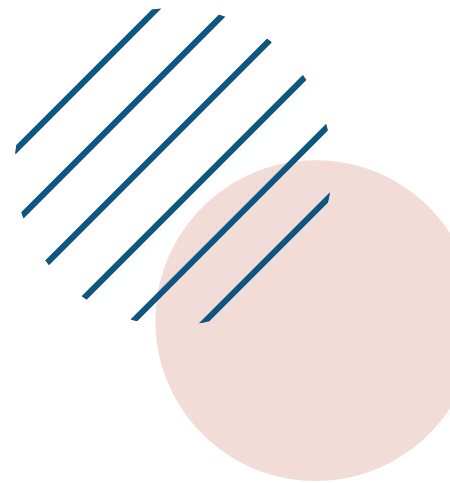
소프트웨어 공급망 보호

2021년 보고서에서는 소프트웨어 공급망을 보호하는 것이 중요한 여러 성과를 달성하는 데 필수적이라는 점을 확인한 바 있습니다.

올해는 소프트웨어 공급망 보안을 설문조사와 보고서의 주요 주제로 선정하여 집중 조명했습니다. [SLSA\(Supply Chain Levels for Software Artifacts\)](#) 프레임워크를 활용하여 소프트웨어 공급망 보안의 개발을 지원하는 기술 관행을 살펴봤습니다. 또한 소프트웨어 공급망 보안과 관련된 태도, 프로세스, 비기술적 관행을 살펴보기 위해 미국 국립표준기술연구소(NIST)의 [SSDF\(Secure Software Development Framework\)](#)를 사용했습니다.

조직의 애플리케이션 개발 보안 관행에서 가장 큰 예측자는 기술이 아닌 문화인 것으로 나타났습니다. 신뢰도가 높고 비난을 삼가며 성과를 중요시하는 문화에서는 신뢰가 낮고 비난을 일삼으며 권력이나 규칙을 중요시하는 문화와 비교해 새로운 보안 관행을 평균 이상으로 채택할 가능성이 1.6배 더 높았습니다. 또한 배포 전 보안 스캔이 취약한 종속 항목을 찾는 데 효과적이어서 프로덕션 코드에서 취약점이 줄어들게 만든다는 것을 시사하는 초기 증거도 확인했습니다.

우수한 애플리케이션 개발 보안 관행의 채택은 추가적인 혜택으로 이어졌습니다. 이러한 보안 관행을 수립하는 데 중점을 두는 팀에서는 개발자 번아웃이 줄어든 반면 보안 관행 수준이 낮은 팀에서는 보안 관행 수준이 높은 팀보다 강한 수준의 번아웃을 겪을 확률이 1.4배 더 높은 것으로 나타났습니다.¹ 보안 관행을 수립하는 데 중점을 두는 팀은 자신의 소속 팀을 다른 사람에게 추천할 가능성이 훨씬 높습니다. 아울러 SLSA 관련 보안 관행은 조직 성과와 소프트웨어 배포 성과를 모두 긍정적으로 예측하지만 이 효과가 충분히 발휘되려면 강력한 지속적 통합 역량이 필요합니다.



¹ 이 통계에서 점수에 대한 표준 편차가 1 이상인 경우 높음으로, -1 이하인 경우 낮음으로 개념화합니다.



조직 성과의 동인

위에 언급된 보안 관행 외에도 조직 성과에 영향을 미치는 주요 변수는 대체로 다음 카테고리에 속합니다.

조직 문화 및 팀 문화

[웨스트럼의](#) 정의에 따르면 신뢰도가 높고 비난을 삼가는 문화에서 조직 성과가 더 높은 경향이 있습니다. 마찬가지로 팀이 자금 및 리더십 후원을 통해 지원을 받는다고 느끼는 조직이 더 높은 성과를 내는 경향이 있습니다. 팀의 안정성과 소속 팀에 대한 긍정적인 인식(팀을 추천할 가능성)도 높은 수준의 조직 성과로 이어지는 경향이 있습니다. 마지막으로, 탄력적인 근무 제도를 갖춘 기업이 높은 수준의 조직 성과를 보이는 경향이 있습니다.

신뢰성

안정성 엔지니어링(예: 명확한 신뢰성 목표, 주요 신뢰성 측정항목 등)과 연관되는 관행과 사람들이 신뢰성 기대치를 충족한다고 보고하는 정도는 조직의 우수한 성과를 강력하게 예측할 수 있는 요인입니다.

클라우드

클라우드 사용이 조직 성과를 예측할 수 있는 요소인 것으로 확인되었습니다. 처음부터 클라우드 기반으로 소프트웨어를 빌드한 기업은 보다 우수한 조직 성과를 내는 경향이 있습니다. 프라이빗 클라우드, 퍼블릭 클라우드, 하이브리드 클라우드 또는 혼합된 형태의 클라우드를 사용하는 경우 온프레미스 서버만 사용할 때와 비교해 조직 성과가 더 높습니다. 여러 퍼블릭 클라우드를 사용하는 조직은 그렇지 않은 조직보다 평균 이상의 조직 성과를 낼 가능성이 1.4배 더 높습니다.

또한 클라우드 사용은 데이터 세트의 다른 요소를 통해서도 조직 성과에 영향을 미치는 것으로 보입니다. 한 가지 예가 공급망 보안인데, 퍼블릭 클라우드를 사용하는 조직은 SLSA 관행을 구현할 가능성이 더 높은 것으로 나타났습니다. 클라우드 제공업체가 빌드 및 배포 자동화와 같은 여러 SLSA 관행을 위한 빌딩 블록을 권장하고 제공하기 때문으로 보입니다.^{2,3} 폭넓게 접근하자면 팀에서 클라우드 플랫폼을 사용할 경우 많은 역량과 관행을 상속하여 궁극적으로 조직 성과가 향상된다고 할 수 있습니다.

²Jung, Sun Jae. "Introduction to Mediation Analysis and Examples of Its Application to Real-world Data." Journal of preventive medicine and public health = Yebang Uihakhoe chí vol. 54,3 (2021): 166-172. doi:10.3961/jpmph.21.069

³ Carrión, Gabriel Cepeda, Christian Nitzl, and José L. Roldán. "Mediation analyses in partial least squares structural equation modeling: Guidelines and empirical examples." Partial least squares path modeling. Springer, Cham, 2017. 173-195.

맥락이 중요

오랫동안 DORA는 팀이 처한 광범위한 맥락에 따라 효과가 달라진다는 점을 고려했습니다. 팀의 특성(프로세스, 강점, 제약 조건, 목표)과 업무가 수행되는 환경을 이해하는 것이 중요하다고 생각합니다. 예를 들어 어떤 상황에서는 유리한 기술적 역량이 다른 상황에서는 해로울 수도 있습니다. 올해는 이렇게 가정된 조건을 상호작용 형태로 명시적으로 모델링하는 데 집중했으며 해당 가설 중 많은 부분이 올해 얻은 데이터로 뒷받침됩니다.

- 높은 소프트웨어 배포 성과는 운영 성과도 높을 때만 조직 성과에 도움이 됩니다. 서비스가 사용자의 신뢰성 기대치를 충족하지 못한다면 신속한 배포는 의미가 퇴색될 수 있습니다.
- SLSA 프레임워크에서 권장하는 것과 유사한 소프트웨어 공급망 보안 제어를 구현하면 지속적 통합이 굳건하게 확립될 때 소프트웨어 배포 성과에 긍정적인 영향을 미칩니다. 지속적 통합 기능이 마련되지 않는다면 소프트웨어 배포 성과와 보안 제어가 상충될 수 있습니다.
- 사이트 안정성 엔지니어링(SRE) 관행이 신뢰성 목표를 달성하기 위한 팀의 역량에 미치는 영향은 비선형적입니다. SRE를 적용한다고 해도 팀이 일정 수준의 SRE 성숙도에 도달하기 전에는 신뢰성에 긍정적인 영향을 미치지 않습니다. 팀이 이 수준의 SRE 성숙도에 도달하기 전에는 SRE와 신뢰성 목표 달성 간의 관계가 인식되지 않습니다. 하지만 팀의 SRE 도입이 증가함에 따라 SRE의 사용이 신뢰성을 분명하게 예측할 수 있는 강력한 요소가 되기 시작하는 변곡점에 도달합니다. 신뢰성 향상은 조직의 성과에 영향을 미칩니다.
- 기술적 역량은 상호 보완적입니다. 지속적 배포와 버전 제어는 높은 수준의 소프트웨어 배포 성과를 촉진하는 역량을 서로 증폭시킵니다. 따라서 지속적 배포, 느슨하게 결합된 아키텍처, 버전 제어, 지속적 통합을 결합해 사용하면 각각의 성과를 합한 것보다 더 나은 소프트웨어 배포 성과를 얻을 수 있습니다.



배포에 영향을 미치는 조건과 팀이 처한 폭넓은 맥락을 이해해야 할 필요성을 토대로, 2021년에 얻었던 다음과 같은 유용한 정보와 유사한 결론에 도달했습니다.

"의미 있는 발전을 이루기 위해 팀에서는 지속적 개선의 철학을 채택해야 합니다. 벤치마크를 사용하여 현재 상태를 평가하고, 연구에서 조사한 역량을 기반으로 제약 조건을 식별하고, 이러한 제약을 완화하기 위한 개선 사항을 실험하세요. 실험에는 성공과 실패가 혼재하지만 팀에서는 두 시나리오 모두에서 얻은 교훈을 바탕으로 의미 있는 조치를 취할 수 있습니다."

실제로, 올해 이 중요한 철학과 일맥상통하는 효과가 확인되었습니다. 지속적으로 개선 필요성을 인식하는 팀은 그렇지 않은 팀보다 조직 성과가 더 높은 경향이 있다는 것입니다.

간단히 말해, 팀은 새로운 상황에 지속적으로 적응하고 여러 소프트웨어 개발 관행을 실험해야 합니다.

전반적으로, 이를 실천하는 팀은 더 나은 조직 성과를 내기 때문이며 항상은 아니지만(한 조직에서는 효과적이라도 다른 조직에서는 효과적이지 않을 수 있음) 대부분의 경우에 그렇습니다. DevOps 관행에 따라 자체 실험에 참여할 때 팀에 효과적인 방법을 모색하는 과정에서 간혹 경험하게 될 실패에 대비하세요.

올해에는 데이터에서 수많은 [놀라운 사실](#)도 발견했습니다. 자세한 내용을 알아보려면 본 보고서를 계속 읽어보세요.

지속적으로 개선 필요성을 인식하는 팀은 그렇지 않은 팀보다 조직 성과가 더 높은 경향이 있습니다.

02

비교 방법



데렉 드벨리스

팀이 업계의 다른 팀과 비교되는 방식이 궁금하신가요? 이 섹션에는 DevOps 성과에 대한 최신 벤치마크 평가가 포함되어 있습니다. 연구에서는 팀이 소프트웨어 시스템을 어떻게 개발, 배포, 운영하는지 조사한 다음 응답자를 DevOps 성과의 가장 일반적인 조합을 캡처하는 클러스터로 분류합니다.

올해는 두 가지의 클러스터링 접근 방법을 사용합니다. 첫 번째는 과거 선례를 따르는 것입니다. 이 클러스터링 접근 방법은 리드 타임, 배포 빈도, 서비스 복원 시간, 변경 실패율과 같은 소프트웨어 배포 성과를 캡처하는 4가지 측정항목을 기반으로 클러스터를 만드는 데 중점을 둡니다. 각 내용에 대한 요약은 뒤에서 다룹니다. 이 접근 방법의 목표는 팀의 현재 성과를 계량화하여 다른 팀의 성과와 비교할 수 있도록 돕는 것입니다.

두 번째 클러스터링 접근 방법에는 운영 성과를 파악하는 데 사용하는 5번째 측정항목인 신뢰성이 포함됩니다. 클러스터 분석에 새 측정항목을 추가하는 이유는 무엇일까요? 이 측정항목의 중요성이 지속적으로 확인되었기 때문입니다. 강력한 운영 성과가 동반되지 않는다면 배포 성과가 조직 성과에 부정적인 영향을 미칠 수 있음을 시사하는 실질적인 증거가 있습니다. 기존의 클러스터링 접근 방법과 달리, 이 방법은 배포 성과와 운영 성과를 달성하기 위해 팀이 수행하는 일반적인 방법을 표현하려는 설명적인 시도입니다. 따라서 어떤 클러스터가 더 나은지 항상 명확하지는 않습니다.

먼저 소프트웨어 배포와 운영 성과를 파악하는 데 사용하는 5가지 측정항목에 대해 간략히 살펴보겠습니다.

소프트웨어 배포 및 운영 성과

조직은 끊임없이 변화하는 업계의 요구를 충족하기 위해 소프트웨어를 빠르고 안정적으로 배포하고 운영해야 합니다. 팀에서 더 빠르게 소프트웨어를 변경할수록 더 빠르게 고객에게 가치를 제공하고 실험을 실행하고 값진 피드백을 받을 수 있습니다. 이 보고서에서는 8년간의 데이터 수집과 연구 결과를 토대로 소프트웨어 배포 성과를 측정하는 4가지 측정항목을 개발하고 검증했습니다. 2018년 이후에는 운영 능력을 파악하기 위한 5번째 측정항목을 포함했습니다.

5가지 측정항목 모두에서 뛰어난 결과를 낸 팀은 탁월한 조직 성과를 보입니다. 이 5가지 측정항목을 **소프트웨어 배포 및 운영(SDO) 성과**라고 합니다. 이러한 측정항목은 시스템 수준의 결과에 초점을 맞추므로 전체 성과를 희생하면서 기능을 서로 결합시키거나 로컬 최적화를 수행하는 결과로 이어질 수 있는 소프트웨어 측정항목 추적의 일반적인 함정을 피하는 데 도움이 됩니다.

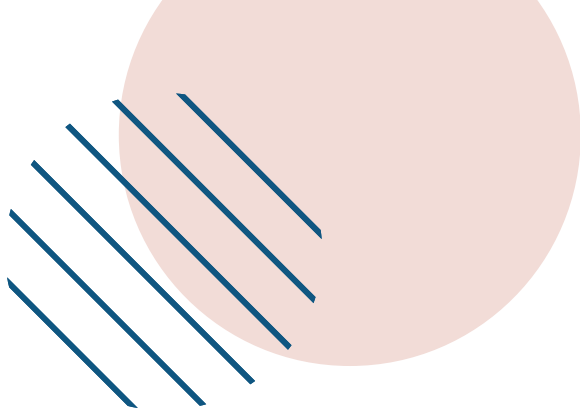


배포 성과와 운영 성과의 5가지 측정항목

소프트웨어 배포 성과의 4가지 측정항목을 처리량과 안정성 측면에서 고려할 수 있습니다. 처리량은 **코드 변경의 리드 타임** (즉, 코드 커밋부터 프로덕션 출시까지의 시간)과 **배포 빈도**를 사용하여 평가합니다. 안정성은 이슈 발생 후 **서비스 복원 시간** 및 **변경 실패율**을 사용하여 평가합니다.

5번째 측정항목은 **운영 성과**를 나타내며 현대적 운영 관행을 측정합니다. **운영 성과**는 서비스가 가용성과 성능과 같은 사용자 기대치를 얼마나 잘 충족하는지를 나타내는 **신뢰성**을 기반으로 합니다. 이전에는 신뢰성보다 가용성을 주로 평가했지만, 가용성은 안정성 엔지니어링에 초점을 두기 때문에 2021년에는 가용성, 지연 시간, 성능, 확장성이 보다 골고루 반영되도록 측정항목에 신뢰성을 포함하는 방향으로 확장했습니다. 특히 응답자에게 신뢰성 목표를 달성하거나 초과하는 역량을 평가해 달라고 요청했습니다. 그 결과 배포 성과의 수준이 다양한 각 팀에서 운영 성과를 우선시할 때 성과가 향상(예: 번아웃 감소)된다는 점을 확인했습니다.





과거의 클러스터링 접근 방법: 클러스터링 배포 성과

2018년부터 사용해온 4가지 클러스터 솔루션을 올해 평가한 결과, 저조한 성과를 보이는 클러스터와 높은 성과를 보이는 클러스터가 데이터에 극명하게 드러나는 것을 확인했습니다. 그러나 전통적으로 중간 성과와 높은 성과를 구분하기 위해 사용하는 두 클러스터는 명확한 경계를 지을 만큼 충분히 차별화되지 않았습니다. 또한 올바른 클러스터 솔루션을 선택하는 데 사용하는 다양한 지표는 적용된 클러스터링

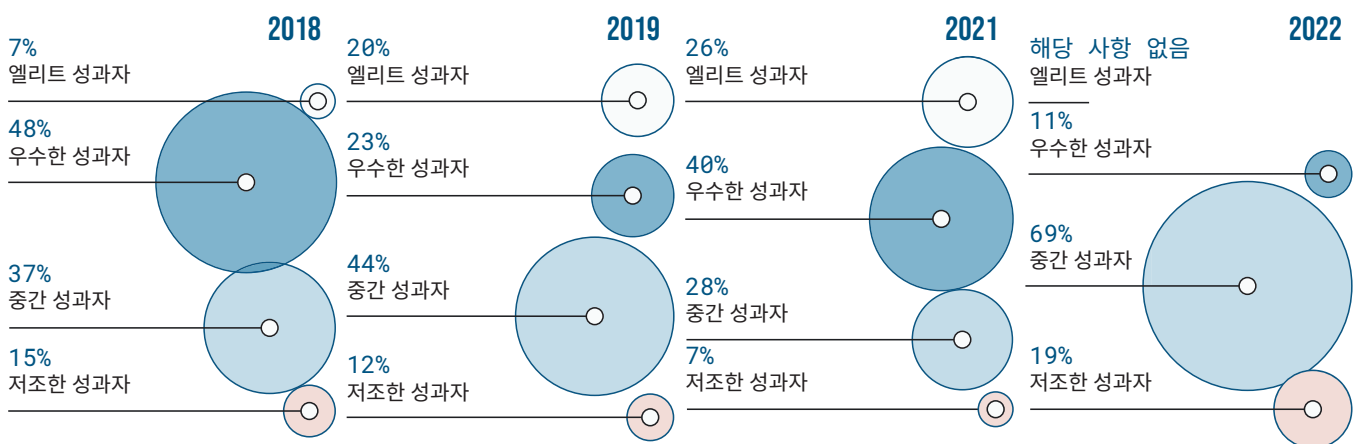
기술에 관계없이 3개의 클러스터가 데이터를 가장 잘 캡처한다는 것을 변함없이 나타냈습니다. 아래 표에서는 각 클러스터의 배포 성과 특징을 설명합니다.

지난해와 눈에 띄는 차이는 올해는 엘리트로 분류할 클러스터가 없다는 점입니다. 올해 성과가 높은 클러스터는 지난해 성과가 높은 클러스터와 엘리트 클러스터를 혼합한 것입니다. 올해 성과가 가장 우수한 클러스터가 지난해 엘리트 클러스터의 특성을 충분히 나타내지 않기 때문에 올해에는 엘리트 클러스터를 생략하기로 결정했습니다.

소프트웨어 배포 성과 측정항목	낮음	중간	높음
배포 빈도 작업하는 주요 애플리케이션 또는 서비스의 경우 조직에서 얼마나 자주 코드를 프로덕션에 배포하거나 최종 사용자에게 출시하나요?	한 달에 한 번에서 6개월에 한 번 사이	일주일에 한 번에서 한 달에 한 번 사이	주문형(하루에 여러 번 배포)
변경 리드 타임 작업 중인 주요 애플리케이션 또는 서비스의 경우 변경 리드 타임(예: 코드 커밋부터 프로덕션 환경에서 코드 실행을 성공하기까지 걸리는 시간)은 얼마인가요?	1개월에서 6개월 사이	일주일에서 1개월 사이	하루에서 일주일 사이
서비스 복원 시간 작업 중인 주요 애플리케이션 또는 서비스의 경우 사용자에게 영향을 미치는 서비스 이슈 또는 결함(예: 예기치 않은 서비스 중단 또는 서비스 손상)이 발생할 때 서비스를 복원하는 데 시간이 보통 얼마나 걸리나요?	일주일에서 1개월 사이	하루에서 일주일 사이	하루 미만
변경 실패율 작업 중인 주요 애플리케이션 또는 서비스의 경우 프로덕션에 대한 변경사항 또는 사용자에게 출시된 변경사항 중 몇 퍼센트가 서비스 저하(예: 서비스 장애 또는 서비스 중단으로 이어짐)를 초래하고 이후에 조치(예: 핫픽스, 롤백, 향후 수정, 패치 적용)가 필요한가요?	46~60%	16~30%	0~15%

이 샘플은 직원이 앞서 나가고 있다고 느끼는 팀이나 조직을 대표하지 않음을 시사합니다. 한 가지 가능한 가설은 (현재 이 가설을 뒷받침할 데이터가 부족하기는 하지만) 관행, 도구, 정보 공유 측면에서 소프트웨어 개발 분야의 혁신이 저하되었다는 것입니다. 팬데믹이 지속되면서 팀과 조직에서 지식과 관행을 공유하기 어려웠기 때문에 발생한 결과일 수 있습니다. 서로에게서 정보를 수집하고 배울 기회가 줄어들어 혁신이 둔화된 것일 수 있습니다. 이러한 결과가 나온 배경이 무엇인지 알아보기 위해 자세히 따져보고자 합니다. 종합적으로 볼 때 올해 성과가 저조하거나, 중간이거나, 높은 클러스터를 지난해와 비교하면 배포 성과가 다소 높은 방향으로 이동하고 있음을 확인할 수 있습니다. 올해의 클러스터는 지난해의 두 클러스터 사이에 있는 것으로

보입니다. 즉, 2022년도에 성과가 높은 클러스터는 2021년도의 성과가 높은 클러스터와 엘리트 클러스터 사이에 있습니다. 2022년도에 성과가 저조한 클러스터는 2021년도의 성과가 저조한 클러스터와 중간 클러스터 사이에 있는 것으로 보입니다. 성과가 저조한 클러스터의 상향 이동은 배포 성과의 상한은 낮아진 반면 하한은 높아졌음을 시사합니다.

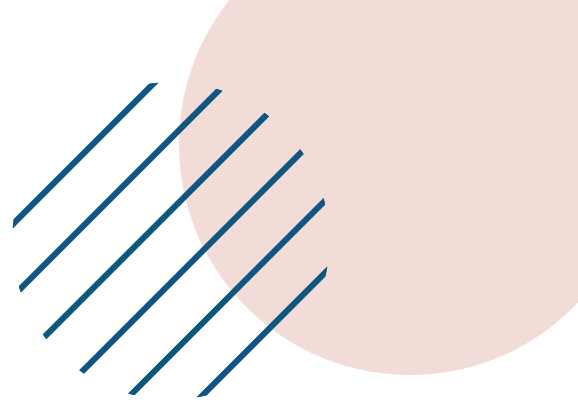


위의 표에서 비율 분류를 보면 우수한 성과자의 비율은 4년만에 최저 수준인 반면 저조한 성과자의 비율은 2021년 7%에서 올해 19%로 급격히 증가한 것을 알 수 있습니다. 올해 응답자의 3분의 2 이상이 중간 클러스터에 속합니다. 우수한 성과자와 엘리트 성과자의 뚜렷한 감소는 올해 응답자 중 다수가 수많은 현대적인 팀에서 목격되는 DevOps 문화를 확립하지 않았거나 확립하는 과정에 있는 조직이나 팀에 속해 있다는 의미일 수 있습니다.

2021년과 2022년의 유사점을 강조하기보다 차이점에 지나치게 집중하고 있는지도 모릅니다. 2021년과 2022년의 클러스터는 우수한 성과자와 저조한 성과자를 크게 구분하는 등 많은 특성을 공유합니다. 예를 들어 우수한 성과자는 저조한 성과자보다 417배 더 많은 배포를 구현하는 것으로 추산됩니다.

¹ 이러한 추정치가 나온 배경은 '방법론' 섹션을 참조하세요.





클러스터링 배포 성과 및 운영 성과

5개 측정항목이 나타내도록 고안된 다음 3가지 카테고리에 대해 클러스터 분석을 수행하기로 했습니다. 3가지 카테고리란 **처리량**(코드 변경의 리드 타임, 배포 빈도의 조합), **안정성**(서비스 복원 시간, 변경 실패율의 조합), **운영 성과**(신뢰성)입니다. 그 이유는 우리 모델에서 운영 성과가 중추적인 역할을 했기 때문입니다. 운영 성과가 우수하지

않은 조직에서는 처리량과 안정성이 조직 성과에 미치는 영향이 미미합니다. 운영 성과를 고려하지 않고 DevOps 성과 배경을 설명하는 것은 전체 그림에서 중요한 부분을 놓치는 것입니다.

데이터를 분석한 결과 4가지 클러스터 솔루션이 도출되었습니다. 다음은 이 4가지 클러스터와 이름을 나타낸 표입니다.

클러스터	안정성		운영 성과	처리량		응답자 비율(%)
	서비스 복원 시간	변경 실패율	신뢰성	리드 타임	배포 빈도	
시작	하루에서 일주일 사이	31~45%	가끔 기대치 충족	일주일에서 1개월 사이	일주일에 한 번에서 한 달에 한 번 사이	28%
진행	1시간 미만	0~15%	대체로 기대치 충족	하루 미만	주문형(하루에 여러 번 배포)	17%
둔화	하루 미만	0~15%	대체로 기대치 충족	일주일에서 1개월 사이	일주일에 한 번에서 한 달에 한 번 사이	34%
중단	1개월에서 6개월 사이	46~60%	대체로 기대치 충족	1개월에서 6개월 사이	한 달에 한 번에서 6개월에 한 번 사이	21%

그러나 같은 클러스터에 속하는 두 팀을 만나보면 매우 다른 인상을 받을 수 있으며 이 스토리가 실제 보이는 모습을 반영하지 않을 수 있습니다. 위의 각 클러스터에 대해 제공하는 스토리는 많은 팀과 협업했던 경험을 활용하여 데이터에 나타나는 패턴을 이해하기 쉽도록 만들려는 시도입니다. 아울러 동일한 팀이라도 서로 다른 시점에 만나보면 같은 클러스터에 배치되지 않을 수도 있습니다. 이는 팀의 성과가 향상되었거나 저하되었기 때문일 수 있습니다. 또는 팀이 애플리케이션 또는 서비스의 현재 상태에 더 적합한 배포 패턴으로 전환했기 때문일 수 있습니다. 예를 들어 애플리케이션 또는 서비스 개발의 초기 시점에 팀은 탐색(시작 클러스터)에 집중했을 수 있지만 틈새를 발견하기 시작하면서 신뢰성(진행 클러스터 또는 둔화 클러스터)으로 초점을 옮길 수 있습니다.

각 클러스터에는 고유한 특성이 있으며 응답에서 상당한 부분을 차지합니다.

시작 클러스터는 모든 측정기준에서 성과가 우수하지도 저조하지도 않습니다. 이 클러스터는 제품, 기능 또는 서비스 개발의 초기 단계에 해당할 수 있습니다. 피드백을 받고, 제품 시장 맞춤 단계를 파악하고, 보다 일반적으로는 탐색에 주력하기 때문에 신뢰성에 덜 집중할 수 있습니다.

진행 클러스터는 신뢰성, 안정성, 처리량 등 모든 특성에서 성과가 우수합니다. 응답자의 17%만이 진행 상태에 해당합니다.

둔화 클러스터에 있는 응답자는 빈번하게 배포하지는 않지만 배포하면 성공할 가능성이 높습니다. 응답자의 3분의 1 이상이 이 클러스터에 속하므로 가장 크고 대표적인 샘플입니다. 이 패턴은 점진적으로 개선되고 있는 팀에 일반적일 가능성이 있지만(고유한 특성은 아님) 팀과 해당 팀의 고객은 애플리케이션 또는 제품의 현재 상태에 대체로 만족합니다.

마지막으로 **중단** 클러스터는 팀과 고객에게 여전히 가치가 있지만 더 이상 활발하게 개발되지 않는 서비스나 애플리케이션에 대해 작업을 진행하는 팀으로 보입니다.

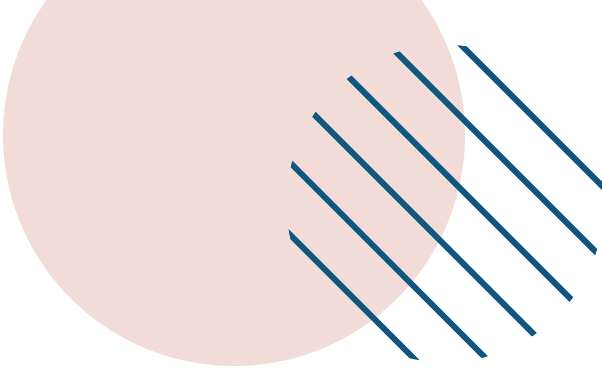
이러한 클러스터는 관행과 기술적 역량이 서로 크게 다릅니다. **진행** 클러스터에서 높은 수준의 소프트웨어 배포 및 운영 성과를 나타낸 것을 감안하여, 관행과 기술적 역량 측면에서 이 클러스터가 다른 클러스터와 어떻게 다른지 살펴봤습니다. 그 결과 **진행** 클러스터는 다른 클러스터에 비해 다음에 더 집중하는 것으로 나타났습니다.

- 느슨하게 결합된 아키텍처: 팀이 시스템을 변경하기 위해 다른 팀에 의존하지 않고 시스템 설계를 대규모로 변경할 수 있는 정도
- 탄력성 제공: 회사가 직원의 근무 제도를 탄력적으로 운영하는 정도
- 버전 제어: 애플리케이션 코드, 시스템 구성, 애플리케이션 구성 등의 변경이 관리되는 방식
- 지속적 통합(CI): 브랜치가 트렁크에 통합되는 빈도
- 지속적 배포(CD): 안전하고, 지속적이며, 효과적으로 변경사항을 프로덕션 환경에 적용하는 데 중점을 둔 역량

흥미롭게도 **진행** 클러스터는 문서화를 덜 중요시하는 경향이 있습니다. 지난해 보고서에서 문서화 관행이 배포 성과와 운영 성과(SDO) 모두에 핵심적인 역할을 한다는 점을 확인한 바 있습니다. **진행** 클러스터는 어떻게 문서화에 관심을 쏟지 않고도 우수한 SDO 성과를 낼까요? 한 가지 이유는, 문서화 외에도 우수한 SDO 성과를 낼 많은 방법이 있다는 것입니다. 또한 **진행** 클러스터는 자체 문서화가 뛰어난 프로세스를 만들기 위해 코드를 지속적으로 리팩터링하고 있으므로 설문조사에서 설명하는 것과 같은 문서의 필요성이 크지 않습니다.

응답자의 가장 많은 비중을 차지하는 **문화** 클러스터는 다른 클러스터보다 클라우드 기반이 약한 경향이 있는 대규모 조직의 응답자로 주로 구성됩니다. 최종 사용자에게 안정적이고 신뢰할 수 있으며 가치 있는 경험을 계속해서 제공하는 견고한 프로세스를 갖춘 성숙도 높은 회사를 상상해 보세요. 이 클러스터는 성과 중심의 생성적 문화를 나타냅니다.² 문화 클러스터의 가장 흥미로운 특징 중 하나는 낮은 처리량과 높은 수준의 긍정적인 업무 문화(웨스트럼의 표현으로는 '생성적' 업무 문화)인데, 이는 흔히 않은 조합입니다. 업무 문화에 비례하여 처리량이 나타나는 것이 더 일반적입니다(문화 수준이 높으면 처리량이 높고, 문화 수준이 낮으면 처리량이 낮음). 처리량과 문화의 관계를 더 잘 파악하기 위한 향후 연구가 진행되기를 기대합니다.

² (웨스트럼)



또한 번아웃, 조직 성과, 무계획적 업무라는 3가지 결과 측면에서 각 클러스터가 어떻게 비교되는지도 살펴봤습니다. 조사 결과는 예상을 빚나갔습니다. **중단** 클러스터는 조직 성과 측면에서 다른 클러스터보다 우수했습니다. 이 클러스터의 특성(낮은 안정성과 낮은 처리량)을 보면 DORA가 이전에 발견한 대부분의 사항과 상충되는 것으로 보입니다. 하지만 무작위성(가능성 매우 높음)으로 인해 이상치가 도출되었다고 단정하는 대신 몇 가지 설득력 있는 설명을 찾아보고자 합니다.

이러한 조사 결과의 분석을 시도할 때는 유념해야 할 중요한 추가 사실이 있습니다. **중단** 클러스터는 우수한 조직 성과를 달성하지만 한편으로는 상당한 대가를 치릅니다. 팀의 번아웃 비율이 가장 높고, 오류에 가장 취약하다고 느끼며, 무계획적 업무로 부담이 가중됩니다. 이러한 결과는 신뢰성을 바탕으로 우수한 조직 성과를 달성할 수도 있지만 속도와 안정성이 뒷받침되지 않는다면 팀이 번아웃과 무계획적 업무라는 대가를 치르게 됨을 알려줍니다.

중단 클러스터가 **진행** 클러스터를 비롯한 다른 클러스터보다 우수한 조직 성과를 보이는 이유를 설명하는 다른 가설도 있습니다. 간략한 목록은 다음과 같습니다.

- 데이터에는 존재하지 않을 수 있으나 4가지 클러스터의 기반이 되는 특성이 있습니다. 예를 들어 조직 규모는 성숙도를 나타내는 적절한 지표가 될 수 있습니다. 소규모 회사는 **진행** 클러스터에 속하는 경향이 있으며 이는 해당 제품이 보다 형성 단계에 있음을 나타냅니다.
- **진행** 클러스터에 속하는 소규모 회사는 이전의 프로세스와 인프라에 덜 얽매일 수 있기 때문에 보다 정교한 DevOps 프로세스를 갖추고 있습니다. 그렇더라도 데이터에 따르면 기술과 큰 연관성이 없을 수 있는 이유로 인해 조직의 규모와 조직 성과 사이에 양의 상관관계가 존재합니다.
- **진행** 클러스터는 웨스트럼이 생성적 문화에서 설명한 원칙을 무시하는 경향이 있습니다. 이것이 조직 성과에 부정적인 영향을 준다는 사실을 종종 확인할 수 있었습니다.
- 각 클러스터의 조직이 정의하는 신뢰성 기대치와 이를 모니터링하는 방법은 서로 다를 수 있습니다. 조직의 성과 목표를 정의하는 방법도 마찬가지입니다.

- **중단** 클러스터는 단기적으로는 높은 조직 성과를 보일 수 있지만 장기적으로는 어떤 성과를 낼지 확실치 않습니다. 번아웃이 이직률 증가로 이어질까요? 프로세스를 확장할 수 있을까요?
- 우리는 다양한 수준에서 질문을 하고 있습니다. 기술적 역량(예를 들어 느슨하게 결합된 아키텍처)은 팀 수준에서 질문하며, 조직 성과와 관련된 질문은 조직 수준에서 질문합니다. 조직은 많은 경우 여러 팀으로 구성되며 응답자가 조직은 잘 운영되더라도 자신이 속한 팀은 그렇지 않다고 인식할 수 있습니다.

또한 **진행** 클러스터는 조직 성과 부문에서 **중단** 클러스터 다음으로 두 번째로 높은 점수를 받았으며 번아웃 및 무계획적 업무가 가장 낮은 수준을 보였습니다. **진행** 및 **둔화** 클러스터에서 입증된 바와 같이 DevOps 철학은 신뢰성이 확보된 상태에서 가장 큰 효과를 발휘합니다.

앞으로도 업계 내부에서 진행되는 변화를 설명할 새로운 방법을 계속 모색할 것입니다. 향후에도 이러한 변화를 이해하기 위한 관련 측정기준에 운영 성과를 포함하고자 합니다. 또한 지나치게 규범적이고 평가적인 클러스터(예: 엘리트)는 피하는 대신 SDO 성과의 공통된 집합을 식별하는 설명적인 시도에 집중할 것입니다.



03

개선 방법



데릭 드벨리스

다양한 성과를 개선하려면 어떻게 해야 할까요?

State of DevOps Report는 팀이 원하는 결과를 얻기 위해 DevOps 관행과 역량에 집중할 수 있도록 도와주는 증거 기반 지침을 제공하기 위해 작성되었습니다. 올해는 보안과 팀이 원하는 결과 집합 모두로 조사를 확대했습니다. 과거에는 소프트웨어 배포 및 운영 성과(SDO)와 조직의 성과 결과에 중점을 두었습니다. 지금도 이러한 내용에 초점을 두기는 하지만 SDO와 조직 성과를 개선하는 수단으로서일 뿐만 아니라 개선 그 자체를 목적으로 번아웃, 소속 팀을 추천할 가능성, 무계획적 업무, 오류 발생 가능성도 살펴보고자 합니다. 이에 따라 올해에는 이러한 결과에 영향을 미치는 것으로 보이는 관행과 역량을 다루게 되었습니다.





올해의 연구 모델은 'DevOps에 획일적인 접근 방법이란 없다'는 DORA의 기본 이론을 보다 잘 반영하도록 그 방향을 전환했습니다. 소속 팀을 추천하기 위해서는 팀을 둘러싼 폭넓은 맥락을 이해하는 과정이 필요하다는 것을 실제로 확인했습니다. 한 팀에 유익한 관행이 다른 팀에는 해로울 수 있습니다. 예를 들어 지속적 배포가 갖추어졌을 때 기술적 역량(예: 느슨하게 결합된 아키텍처, 트렁크 기반 개발, 버전 제어, 지속적 통합)이 소프트웨어 배포 성과에 더욱 분명하게 긍정적인 영향을 미친다는 가설을 오랫동안 유지했습니다. 올해는 이 가설과 기타 상호작용을 명시적으로 모델링했습니다. 목표는 '무엇이 무엇에 영향을 미치는가?'라는 단순한 질문에서 벗어나 '어떤 조건에서 이러한 효과가 나타나거나 강화되거나 약화되는가?'라는 질문을 포함하도록 이해를 높이는 것입니다. 이 모든 조건을 이해하는 것은 복잡하고 어려운 과정으로 입증되었지만 초기 결과 중 일부를 공유하게 되어 기쁩니다.

올해와 지난해의 연구 모델은 [Google Cloud 웹사이트](#)에서 확인할 수 있습니다.

4가지 핵심 측정항목 활용

DORA 측정항목이 개발 성과와 운영 성과를 어떻게 개선할까요? Liberty Mutual Insurance의 소프트웨어 엔지니어 교차 기능팀은 DORA의 ['4가지 핵심' 측정항목](#)을 사용하여 성과를 정기적으로 검토합니다. 예를 들어 Liberty Mutual의 수석 스크럼 마스터인 제나 데일리는 팀이 병목 현상을 파악하고, 테스트 기반 개발 접근 방법으로 전환하며, 전반적인 성과 향상을 실현하는 데 도움이 되도록 DORA의 연구를 활용했다고 말했습니다.

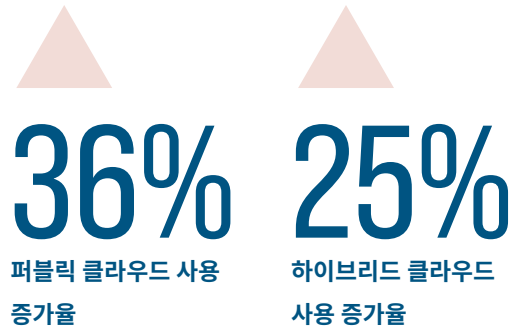
데이터와 DORA 측정항목을 활용하여 소프트웨어의 품질과 배포를 개선하는 Liberty Mutual의 접근 방법을 소개한 최근 [Tomorrow Talks](#)에서 자세한 내용을 알아보세요.



에릭 맥스웰

클라우드

지난 몇 년 동안 형성된 모멘텀을 바탕으로 클라우드 컴퓨팅 사용이 계속해서 가속화되고 있습니다. 실제로 여러 클라우드를 포함하여 퍼블릭 클라우드를 사용한다고 보고하는 사람의 비율은 2021년의 56%에서 현재 76%로 상승했습니다. 프라이빗 클라우드를 사용하지 않는 사람을 포함해 클라우드를 전혀 사용하지 않는다고 보고한 사람의 비율은 지난해 21%에서 10.5%로 하락했습니다. 여러 퍼블릭 클라우드를 사용하는 응답자 비율은 21%에서 26%로 증가했으며 하이브리드 클라우드는 25% 증가해 42.5%에 이르렀습니다. 또한 프라이빗 클라우드를 사용한다는 응답은 지난해의 29%에서 소폭 증가한 32.5%를 기록했습니다.



클라우드 컴퓨팅을 사용하면 전반적인 조직 성과에 긍정적인 영향을 미칩니다. 클라우드를 사용한 응답자는 클라우드를 사용하지 않는 응답자보다 조직 성과 목표를 **초과 달성할 가능성이 14% 더 높았습니다.**

지난 몇 년 동안 확인했고 이 보고서에서 계속해서 검증하고 있듯, 클라우드 컴퓨팅의 사용은 전반적인 조직 성과에 긍정적인 영향을 미칩니다. 클라우드를 사용한 응답자는 클라우드를 사용하지 않는 응답자보다 조직 성과 목표를 **초과 달성할 가능성이 14% 더 높았습니다.** Google Cloud의 연구에 따르면 팀은 클라우드 컴퓨팅을 통해 소프트웨어 공급망 보안과 신뢰성 같은 측면에서 탁월한 성과를 거둘 수 있으며 이는 곧 조직의 성과 향상으로 이어집니다.

놀랍게도 모든 유형의 클라우드(퍼블릭, 프라이빗, 하이브리드, 멀티) 사용자가 변경 실패율과 부정적인 연관성을 보였습니다. 이는 곧 변경 실패율 증가를 필요합니다. 그 이유를

추측하기보다는 향후 연구를 통해 면밀히 조사할 것입니다. 그러나 몇 가지 예외를 제외하면, 클라우드 기반 애플리케이션(원래 클라우드용으로 설계 및 구성된 애플리케이션)의 사용은 설문조사한 모든 항목에서 긍정적인 신호를 나타냈습니다.

퍼블릭 클라우드인지 프라이빗 클라우드인지에 관계없이 클라우드 컴퓨팅 플랫폼의 사용은 문화 및 업무 환경 결과(예: 생성적 문화, 낮은 번아웃, 높은 안정성, 높은 직원 만족도)에 긍정적으로 기여합니다. 클라우드 사용자는 이러한 문화적 결과에 대해 16% 더 높은 점수를 받았습니다.

전년 대비 계속해서 빠르게 증가하는 클라우드 사용률

	2022	2021년 대비 변동률(%)
하이브리드 클라우드	42.47%	25%
퍼블릭/여러 퍼블릭	76.08%	36%
프라이빗 클라우드	32.55%	12%
클라우드 사용 안 함	10.55%	-50%

하이브리드 클라우드와 멀티 클라우드(및 프라이빗 클라우드)의 사용은 응답자가 높은 수준의 신뢰성을 갖추지 않는다면 소프트웨어 배포 성과 지표(MTTR, 리드 타임, 배포 빈도)에 부정적인 영향을 미치는 것으로 보입니다.

조직 성과를 주도하는 하이브리드 및 멀티 클라우드

하이브리드 클라우드와 여러 퍼블릭 클라우드의 사용이 조직에 긍정적인 영향을 미친다는 강력한 신호가 계속해서 확인되고 있습니다. 여러 클라우드를 사용한 실무자는 클라우드를 사용하지 않는 사용자에 비해 조직 성과가 1.4배 더 높은 것으로 나타났습니다. 하지만 하이브리드 클라우드와 멀티 클라우드(및 프라이빗 클라우드)의 사용은 응답자가 높은 수준의 신뢰성을 갖추지 않는다면 여러 소프트웨어 배포 성과 지표(MTTR, 리드 타임, 배포 빈도)에 부정적인 영향을 미치는 것으로 보입니다. 이러한 결과는

견고한 SRE 관행의 중요성과 소프트웨어 배포에서 신뢰성이 기여하는 역할을 잘 보여줍니다.

2021년에는 응답자에게 여러 퍼블릭 클라우드를 활용하는 주된 이유를 질문한 반면 2022년에는 응답자에게 여러 클라우드 제공업체를 활용하여 얻을 수 있는 모든 이점에 대해 물었습니다. 가용성은 가장 많이 보고된 이점 중 하나였으며 이는 Google Cloud가 신뢰성과 관련하여 업계에서 확인해온 관심 및 주목도와 일치합니다. 서비스를 사용할 수 없다면 신뢰할 수 있는 서비스를 제공할 수 없습니다. 실무자의 50% 이상이 다양한 클라우드 제공업체의 고유한 이점을 활용한다고 보고했습니다.

여러 클라우드 제공업체를 채택함으로써 얻을 수 있는 이점

가용성	62.61%
각 제공업체의 고유한 이점 활용	51.59%
신뢰가 여러 제공업체에 분산됨	47.54%
재해 복구	43.48%
법적 규정 준수	37.97%
협상 전략 또는 조달 요구사항	19.13%
기타	4.06%

> 50%

여러 클라우드 제공업체
사용을 보고한 응답자 비율

클라우드 컴퓨팅의 5가지 특성을 활용하는 것은 조직 성과를 향상하는 긴 인과관계의 중요한 출발점입니다.

클라우드 컴퓨팅의 5가지 특성

이번 설문에서는 이전의 연구 접근 방법을 유지하면서 응답자가 단순히 클라우드 컴퓨팅 기술을 사용하고 있는지 여부뿐만 아니라 사용하고 있는 방식을 파악하고자 했습니다. 이를 위해 미국 국립표준기술연구소(NIST)에서 정의한 클라우드 컴퓨팅의 5가지 필수 특성에 대해 질문했습니다.

주문형 셀프서비스 - 소비자는 제공업체 측에 인적 상호작용을 요구하지 않고도 필요에 따라 자동으로 컴퓨팅 리소스를 프로비저닝할 수 있습니다.

광범위한 네트워크 액세스 - 기능을 널리 사용할 수 있으며 소비자가 휴대폰, 태블릿, 노트북, 워크스테이션 등 여러 클라이언트를 통해 액세스할 수 있습니다.

리소스 풀링 - 제공업체 리소스는 멀티 테넌트 모델에서 풀링되고, 물리적 리소스와 가상 리소스가 수요에 따라동적으로 할당 및 재할당됩니다.

일반적으로 고객은 제공된 리소스의 정확한 위치를 직접 제어할 수 없지만 국가, 주 또는 데이터 센터와 같은 더 높은 수준의 추상화에서 위치를 지정할 수 있습니다.

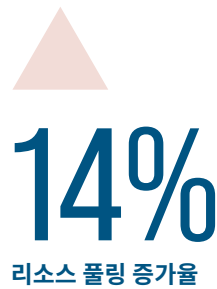
빠른 탄력성 - 기능을 탄력적인 방식으로 프로비저닝 및 출시하여 수요에 따라 빠르게 수평으로 확장 또는 축소할 수 있습니다. 프로비저닝에 사용할 수 있는 소비자 기능은 무제한인 것으로 보이며 언제든지 수량에 관계없이 사용할 수 있습니다.

측정되는 서비스 - 클라우드 시스템은 스토리지, 처리, 대역폭, 활성 사용자 계정과 같은 서비스 유형에 적합한 추상화 수준에서 한도 측정 기능을 활용하여 리소스 사용량을 자동으로 제어하고 최적화합니다. 리소스 사용량을 모니터링, 제어, 보고하여 투명성을 확보할 수 있습니다.

이 보고서는 조직에서 이러한 5가지 특성을 채택할 경우 소프트웨어 배포와 운영 성과에 긍정적인 영향을 미친다는 지난 3년간의 DORA 연구 결과를 입증합니다. 또한 이러한 특성이 조직에 긍정적인 방식으로 영향을 미치는 프로세스를 확립하여 조직 성과를 향상한다는 것을 발견했습니다. 클라우드 컴퓨팅의 5가지 특성을 보여주는 것은 조직 성과의 향상으로 이어지는 긴 여정의 첫 번째 단계입니다.

2022년 연구에서는 팀이 클라우드 컴퓨팅의 차별화 요소를 점점 더 많이 활용하고 있음을 알 수 있습니다. 클라우드 컴퓨팅의 5가지 특성에 대한 채택률이 4년 연속 증가하고 있는 것으로 나타납니다. 리소스 풀링은 14% 증가하여 가장 큰 폭의 성장을 보였고 지난해 두 번째로 많이 사용된 빠른 탄력성은 5%로 증가율이 가장 낮았습니다.

NIST	2021	2022	변경 비율
광범위한 네트워크 액세스	74%	80%	8
빠른 탄력성	77%	81%	5
주문형 셀프서비스	73%	78%	7
측정되는 서비스	78%	83%	7
리소스 풀링	73%	83%	14





데이브 스탠크

SRE 및 DevOps

성공적인 기술팀은 조직에 (심지어 양질의) 코드를 제공하는 것 이상의 기여를 합니다. 또한 기술팀에서 제공하는 서비스가 가용성과 성능을 유지하며 시간이 지남에 따라 사용자의 기대와 일치하도록 지원합니다. 신뢰성은 팀이 이러한 약속을 얼마나 잘 지키는지를 평가하는 다면적인 척도이며 올해 연구에서도 신뢰성을 소프트웨어 배포 및 운영의 한 요소로서 계속 검토했습니다.

사이트 안정성 엔지니어링(SRE)은 Google에서 시작하여 현재 많은 조직에서 적용하고 있는 영향력 있는 운영 접근 방법입니다. SRE는 경험적 학습, 여러 직종의 팀 간 협업, 광범위한 자동화 활용, 서비스 수준 목표(SLO)를 포함한 측정 기술 사용을 우선시합니다. 다른 현대적 운영 관행에서도 유사한 방법을 사용하지만 적용하는 이름이 다릅니다. 따라서 이러한 관행의 범위를 최대한 객관적으로 평가하기 위해 응답자에게 제공하는 설문조사 텍스트에서 중립적이고 설명적인 언어를 사용합니다. 또한 팀이 신뢰성 목표를 달성할 수 있는 정도인 안정성 엔지니어링의 결과에 대한 데이터를 수집합니다. 입력과 출력(SRE 관행 및 신뢰성 결과) 모두 다른 DevOps 역량과 함께 예측 모델에 반영됩니다.

신뢰성이 필수

설문조사를 실시한 팀에서는 SRE를 보편적으로 채택하고 있으며 응답자의 대다수가 우리가 질문한 관행 중 하나 이상을 사용합니다. 많은 팀의 데이터를 통해 신뢰성, 소프트웨어 배포, 성과 간의 미묘한 관계를 알 수 있습니다. 신뢰성이 낮은 상태에서는 소프트웨어 배포 성과로 조직의 성공을 예측할 수 없습니다. 그러나 신뢰성이 향상되면 소프트웨어 배포가 비즈니스 성공에 긍정적인 영향을 미치는 것으로 확인됩니다.

**신뢰성이 없는 상태에서는
소프트웨어 배포 성과로 조직의
성공을 예측할 수 없습니다.**

SRE에 대한 투자는 신뢰성 향상으로 이어지지만, 이는 도입 기준에 도달한 경우에만 그렇습니다.

이 현상은 SRE '오류 예산' 프레임워크를 사용할 때도 동일하게 나타납니다. 서비스의 신뢰성이 떨어진다면 사용자는 취약한 컨텍스트에 코드를 신속히 푸시한다고 해도 이점을 누릴 수 없습니다.

사이트 안정성 엔지니어가 오랫동안 주장해 온 것처럼 신뢰성은 모든 제품에서 가장 중요한 '특성'입니다. 이번 연구는 사용자와의 약속을 지키는 것이 향상된 소프트웨어 배포를 통해 조직을 이롭게 하기 위한 필수 조건이라는 관찰 결과를 뒷받침합니다.

J 곡선 확인

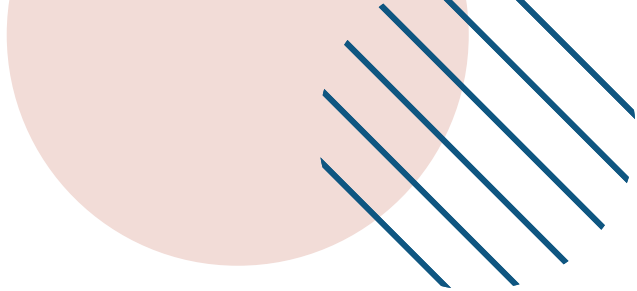
신뢰성을 달성하는 과정에서 직면하는 문제는 무엇인가요? O'Reilly 간행물인 'Enterprise Roadmap to SRE'에서 DORA 설문조사 참여자인 제임스 브룩뱅크와 스티브 맥기는 기존 조직에서 SRE를 구현한 경험을 돌아보고 '변화의 J 곡선을 확인' 하라고 권합니다. 앞서 2018 State of DevOps Report에서 설명한 바 있는 'J 곡선'은 조직의 혁신이 초기에 성공을 거둔 이후 수익이 감소하거나 심지어 회귀하는

기간이 이어지는 현상입니다. 하지만 이러한 힘든 과정을 버텨낸 조직은 대개 새롭고 지속적인 수준의 높은 성취를 경험합니다.

올해는 조사를 진행한 기술팀 전반에서 J 곡선 패턴이 나타났습니다. 팀이 따르는 안정성 엔지니어링 관행의 수가 적은 경우(팀이 SRE 도입 과정의 초기 단계에 있음을 의미함) 이러한 관행으로는 신뢰성 향상을 예측하기가 어렵습니다. 그러나 팀이 더 많은 SRE를 도입함에 따라 SRE의 사용과 신뢰성, 나아가 조직 성과 간에 명확한 관계가 나타나는 변곡점에 도달하게 됩니다.

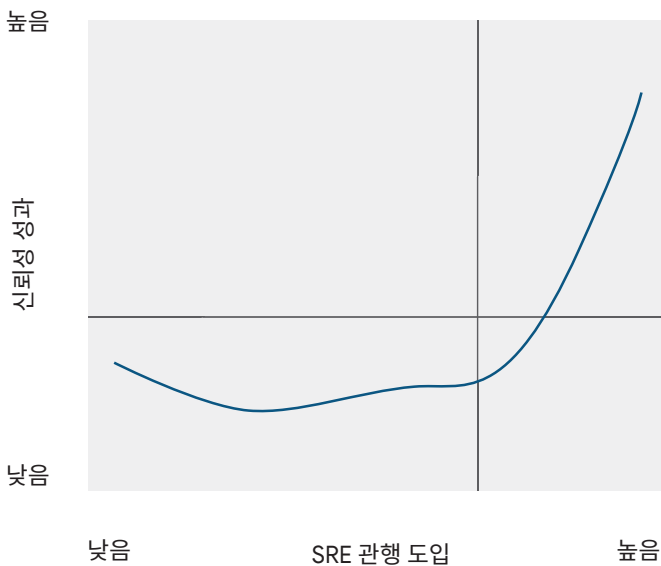


¹ <https://sre.google/resources/practices-and-processes/enterprise-roadmap-to-sre/>



신뢰할 수 있는 팀이 신뢰할 수 있는 서비스를 만듭니다. 생성적 팀 문화는 신뢰성 향상으로 이어집니다.

SRE 관행을 향한 여정의 초기 단계에 있는 팀은 이 과정에서 난관에 대비해야 합니다. 문화, 프로세스, 도구가 모두 새로운 기본 원칙에 맞춰 재편되기 때문에 긴 여정이 될 수 있습니다. 하지만 시간과 지속적인 투자가 병행된다면 성공할 가능성이 높다는 것을 확인할 수 있습니다.



SRE 도입 초기 단계를 지나 관행을 유지하는 팀은 신뢰성이 더욱 향상되는 것으로 나타남

인력, 프로세스, 도구에 대한 투자

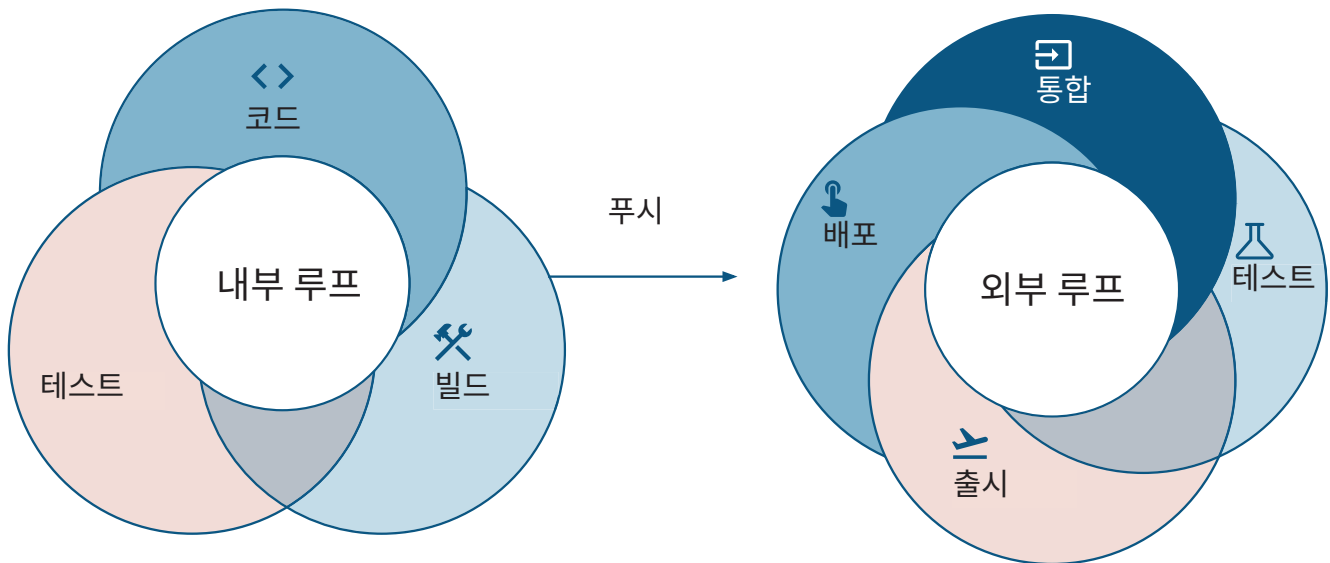
신뢰성은 인간의 노력으로 얻을 수 있는 것이며 SRE 접근 방법이 다방면에서 이를 보여주는 사례입니다. SRE의 핵심 원칙 중 하나는 내부 모니터링 데이터가 아닌 사용자 인식이야말로 신뢰성의 진정한 척도라는 것입니다. 따라서 긍정적인 팀 에너지가 신뢰성을 좌우한다는 것은 그리 놀라운 일이 아닙니다. 신뢰와 협업을 보여주는 '생성적' 문화를 가진 팀이 SRE를 실행할 가능성이 더 높고 우수한 신뢰성을 달성할 가능성이 더 높다는 것을 발견했습니다. 또한 시간에 따른 팀원 구성이 변함없는 안정적인 팀이 사용자 대상 서비스에 더 높은 신뢰성을 제공합니다. DevOps 전체와 마찬가지로 안정성 엔지니어링 작업은 프로세스와 도구를 이용한 인간의 작업으로 보완됩니다. 클라우드 컴퓨팅 및 지속적 통합 사용과 같은 관행은 신뢰성 향상으로 이어집니다.

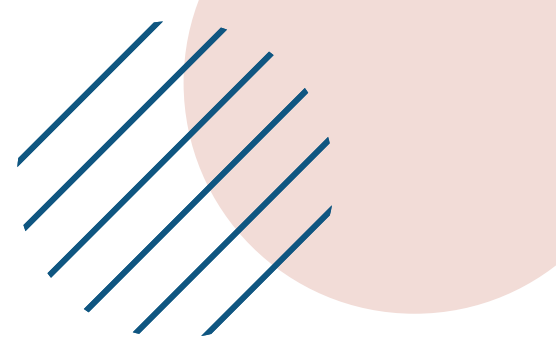


에릭 맥스웰

기술 DevOps 역량

올해는 여러 기술적 관행에 의해 좌우되는 성과를 이해하기 위해 다양한 기술적 역량을 살펴보았습니다. 소프트웨어 개발의 두 가지 광범위한 단계를 고려했는데 바로 코딩, 테스트, 버전 제어 푸시와 같은 개발자 작업으로 구성된 '내부 루프'와 코드 병합, 자동화된 코드 검토, 테스트 실행, 배포, 출시와 같은 활동을 포함하는 '외부 루프'입니다.





신뢰성 목표를 충족하는 우수한 성과자는 CI를 사용할 가능성이 1.4배 더 높습니다.

연구에 따르면 내부 및 외부 루프 개발에 뛰어난 회사는 코드를 더 빠르고 더 높은 수준의 신뢰성으로 제공할 수 있습니다. 우수한 성과에 가장 많은 영향을 주는 기능은 버전 제어, 지속적 통합, 지속적 배포, 느슨하게 결합된 아키텍처입니다.

신뢰성 목표를 충족하는 우수한 성과자의 특징:

33% 버전 제어를 사용할 가능성이 더 높음

39% 지속적 통합을 실행할 가능성이 더 높음

46% 지속적 배포를 실행할 가능성이 더 높음

40% 느슨하게 결합된 아키텍처를 기반으로 하는 시스템을 보유할 가능성이 더 높음

실제로, 위의 모든 역량을 평균 이상으로 사용하는 응답자는 이러한 기술적 역량을 사용하지 않는 응답자보다 **조직 성과가 3.8배 더 높습니다.**

지속적 통합

대개 CI라고도 하는 지속적 통합은 아티팩트를 자동으로 빌드하고, 코드를 배포할 준비가 되었는지 평가하기 위해 모든 코드 커밋에 대해 일련의 자동화된 테스트를 실행하는 외부 루프 개발 프로세스의 일부입니다. 이 프로세스는 개발자에게 빠르고 자동화된 피드백을 제공하여 더 높은 수준의 확신을 가지고 작업할 수 있도록 합니다. CI는 개발자의 워크스테이션에서 프로덕션 환경으로 코드를 가져오는데 핵심적인 부분입니다. 지난해 연구 결과와 마찬가지로 CI는 배포 실적을 좌우하는 것으로 나타났습니다. **신뢰성 목표를 충족하는 우수한 성과자는 다른 사람보다 CI를 사용할 가능성이 1.4배 더 높습니다.**

올해는 외부 루프 개발 프로세스의 다른 부분인 지속적 배포에 대해 더 심도 있게 다루었으며 이 내용은 보고서 후반부에서 설명합니다. 우선, 지속적인 통합에 대한 보완 구성요소인 트렁크 기반 개발을 살펴보겠습니다.

트렁크 기반 개발

트렁크 기반 개발은 지속적으로 코드를 트렁크에 병합하고 수명이 긴 기능 브랜치를 피하는 관행입니다. 이 관행은 지속적 통합을 보완하는 것으로 간주되며 몇 년 동안 소프트웨어 배포 속도를 가속화하는 것으로 나타났습니다.

올해는 업무 경력 기간에 관한 인구통계 변화가 나타남에 따라 트렁크 기반 개발을 구현할 때 경험이 중요하다는 것을 알 수 있었습니다. 지난해에는 응답자의 40%가 16년 이상 근무했다고 밝혔지만 올해는 이러한 응답 비율이 13%에 불과했습니다. '배포에 영향을 주는 요인'이라는 주제를 계속 검증한 결과, 전반적으로 경력이 짧은 그룹은 트렁크 기반 개발과 관련하여 다음과 같이 그다지 긍정적이지 않은 결과를 내는 것으로 확인됩니다.

- ▼ 감소 전반적인 소프트웨어 배포 성과
- ▲ 증가 무계획적인 업무량
- ▲ 증가 오류 발생 가능성
- ▲ 증가 변경 실패율

트렁크 기반 개발을 사용하며 경력이 16년 이상인 개인은 이 관행의 이점을 실현하며 다음과 같은 효과를 보고 있습니다.

- ▲ 증가 전반적인 소프트웨어 배포 성과
- ▼ 감소 무계획적인 업무량
- ▼ 감소 오류 발생 가능성
- ▼ 감소 변경 실패율

이는 트렁크 기반 개발을 성공적으로 구현하는 데 필요한 추가 관행 때문일 수 있습니다. 손상된 트렁크를 절대 방치하지 않는 규칙을 엄격하게 지키지 않거나, 게이트드 코드 브랜치를 사용하지 않거나, 트렁크를 손상시키는 코드의 자동 롤백을 사용하지 않는 팀은 트렁크를 개발할 때 분명히 어려움을 겪을 수밖에 없습니다.

하지만 트렁크 기반 개발의 존재는 전반적인 조직 성과에 긍정적인 영향을 미칩니다.



프랭크 수

지속적 배포

지속적 배포(CD)는 다음과 같은 소프트웨어 개발 관행입니다.

1. 팀이 언제든지 프로덕션 또는 최종 사용자에게 소프트웨어를 배포할 수 있습니다.
2. 새로운 기능에 대해 작업할 때를 포함하여 수명 주기 내내 소프트웨어가 배포 가능한 상태인지 확인합니다.
3. 팀이 시스템의 품질과 배포 가능성을 확인할 수 있도록 신속한 피드백 루프를 설정하고 배포를 방해하는 문제를 해결하는 데 우선순위를 둡니다.

지속적 배포가 반드시 모든 소프트웨어 빌드가 자동으로 배포되는 관행인 지속적 배포를 의미하는 것은 아닙니다. 지속적 배포에서는 소프트웨어 빌드를 언제든지 배포할 수 있기만 하면 됩니다.

지난해에는 팀이 CD를 실행할 가능성을 예측하는 기술 DevOps 역량을 조사했으며 느슨하게 결합된 아키텍처 및 지속적 테스트/통합과 같은 요소가 가장 강력한 예측 요인에 속한다는 사실을 발견했습니다. 올해는 CD 사용을 촉진하는 요인 외에도 개발 결과에 대한 CD 단독 효과와 다른 DevOps 역량과의 상호작용을 분석하고 확인했습니다.

CD가 소프트웨어 배포 성과에 미치는 영향

지난해 조사 결과와 마찬가지로 CD 사용은 단독으로 또는 다른 DevOps 역량과 함께 소프트웨어 배포 성과의 향상으로 이어지는 예측 요인입니다. CD에서 더 높은 평가를 받은 팀은 프로덕션에 코드를 배포하는 빈도가 더 높고 변경 및 서비스 복원을 위한 리드 타임이 더 짧을 가능성이 높습니다.

버전 제어와 지속적 배포를 함께 사용하는 팀은 한 가지에만 집중하는 팀보다 소프트웨어 배포 성과가 높을 가능성이 2.5배 더 높습니다.

또한 응답자의 소속 팀에서도 버전 제어 관행을 채택할 때 더 높은 소프트웨어 배포 성과를 보고할 가능성이 2.5배 높습니다.

CD로 인한 무계획적 업무 증가 가능성

데이터에 따르면 지속적 배포로 인해 개발자는 재작업이나 무계획적 업무에 더 많은 시간을 할애하게 됩니다. 이 결과를 설명하는 한 가지 가설은 더 긴밀한 피드백 루프가 있을 때 개발자가 애플리케이션을 반복적으로 빌드할 가능성이 더 높다는 것입니다. 따라서 개발자는 일부 반복적인 변경을 시스템의 동일한 부분에서 진행되는 무계획적인 업무로 볼 수 있습니다. 이 업무는 동시에 계획되지 않을 수 있으며 이전 배포 시 받은 피드백에 의해 진행될 수 있습니다.

기술 관행과 CD

연구에 따르면 광범위한 기술적 역량이 CD를 지원하는 것이 정기적으로 확인되었습니다. 올해는 이러한 개별적인 역량 중 일부를 CD와 함께 사용할 때 어떤 일이 발생하는지 살펴보았습니다. 그 결과 트렁크 기반 개발 및 느슨하게 결합된 아키텍처를 CD와 함께 사용하면 팀의 성과에 부정적인 영향을 미칠 수 있는 것으로 나타났습니다. 예를 들어 느슨하게 결합된 아키텍처와 CD를 함께 도입한 팀은 CD만 도입한 팀보다 평균 이상의 오류 발생 가능성(즉, 제품 중단, 보안 취약점, 서비스에 발생하는 상당한 성능 저하)을 예상할 가능성이 43% 더 높다는 증거를 확인했습니다. 이러한 효과는 더 많은 연구가 필요하며 개선 중인 팀에 마찰 요소가 잠재되어 있음을 가리킵니다. 이 마찰은 혁신의 J 곡선과 관련이 있을 수 있습니다. 팀은 초기에 개선을 이루지만 달성하기 쉬운 목표를 이룬 후에는 정체됩니다. 잠재력을 최대한 발휘하려면 개선하려는 노력이 필요합니다. CD와 같은 역량을 개선할 때는 팀과 전반적인 성과에 미치는 영향을 반드시 지켜봐야 합니다.



데이비드 팔리

느슨하게 결합된 아키텍처

느슨하게 결합된 시스템은 팀과 조직의 효율성에 중요합니다. 이것은 클라우드 또는 마이크로서비스 기반 시스템에만 적용되는 것이 아니라 조직의 변화 능력과 관련이 있습니다. 조직이 소프트웨어를 안전하고 확신을 갖고 변경할 수 있는 용이성은 소프트웨어의 품질 지표입니다.

느슨하게 결합된 아키텍처를 갖춘 팀은 다음을 수행할 수 있습니다.

- 시스템 변경을 위해 다른 팀에 의존할 필요 없이 시스템 설계를 대규모로 변경
- 더 저렴한 조정 비용으로 독립적인 주문형 테스트를 통해 더 빠른 피드백 확보
- 다운타임이 거의 없이 코드 배포

올해 보고서에서는 응답자에게 구축하는 소프트웨어가 느슨하게 결합된 아키텍처를 기반으로 하는지 여부를 설명하도록 요청했습니다. 그러자 흥미로운 결과가 도출되었으며, 느슨하게 결합된 아키텍처의 존재와 여러 측정기준의 팀 성과 사이에 대부분 긍정적인 다양한 연관성이 나타났습니다.



느슨하게 결합된 아키텍처의 이점

느슨하게 결합된 아키텍처로 소프트웨어를 구축하는 데 중점을 둔 팀은 안정성, 신뢰성, 처리량 전반에 걸쳐 강력한 성과를 발휘할 수 있는 더 나은 위치에 있습니다. 또한 이러한 팀원들은 지인이나 동료에게 자신의 직장을 추천할 가능성이 더 높습니다.

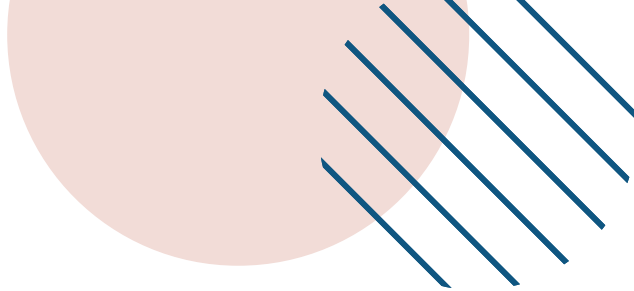
클라우드에 배포하고, 마이크로서비스 아키텍처 접근 방법을 채택하며, 수백 개의 서비스를 관리하는 팀에서 느슨하게 결합된 아키텍처를 사용하는 소프트웨어를 보는 것이 일반적입니다.

느슨하게 결합된 아키텍처로 소프트웨어를 구축하는 데 중점을 둔 팀은 안정성, 신뢰성, 처리량 전반에 걸쳐 강력한 성과를 발휘할 수 있는 더 나은 위치에 있습니다.

그러나 느슨한 결합은 시스템의 서비스 수를 측정하는 단순한 척도 그 이상입니다. 느슨하게 결합된 아키텍처의 구성요소는 독립적으로 배포할 수 있습니다. 이러한 독립성을 통해 팀은 팀 간에 비용이 많이 드는 조정 오버헤드 없이 서비스를 개발, 테스트, 배포할 수 있습니다.

현실 세계에서 느슨한 결합은 하나의 아키텍처 스타일에 제한되지 않습니다. 기본적으로 변경사항이 다른 부분에 영향을 주지 않고 시스템의 한 부분을 변경할 수 있는 기능입니다. 이를 통해 조직은 업무를 분담할 수 있으므로 개별 팀은 다른 팀과 조정하지 않고도 업무를 진행할 수 있습니다.

경험상 소프트웨어를 배포하기 전에 소프트웨어에 대한 신뢰를 쌓기 위한 방법으로 다른 서비스와의 긴밀한 통합 테스트를 수행해야 하는 팀은 아직 느슨한 결합을 구현하지 못한 것입니다. 이러한 팀은 느슨한 결합을 위해 시스템 간의 인터페이스와 격리를 개선하는 것이 좋습니다. 인터페이스와 격리를 개선하는 효과적인 방법 중 하나는 서비스 및 구성요소의 '테스트 가능성'을 개선하는 것입니다. 설계에 따라 서비스를 격리된 상태에서 테스트할 수 있다면 해당 인터페이스는 정의에 따라 느슨하게 결합된 것입니다.



또한 느슨하게 결합된 아키텍처를 사용하는 응집력 있고 안정적인 팀이 지속적인 개선을 장려하고 지원하는 소프트웨어 개발 관행을 사용할 가능성이 더 높은 것으로 나타났습니다. 예를 들어 업무 우선순위를 지정하기 위해 신뢰성 목표를 설정하거나, 증거를 기반으로 신뢰성 목표를 수정하기 위해 정기적인 검토를 수행하는 등의 SRE 관행은 둘 다 느슨하게 결합된 아키텍처를 지원합니다.

또한 느슨하게 결합된 아키텍처를 사용하면 조직에서 직원을 더 쉽게 늘릴 수 있습니다. 다른 팀과 조정할 필요가 없는 독립적인 팀이 독립적으로 팀 규모를 더 자유롭게 확대할 수 있기 때문입니다.

간단히 말해, 소프트웨어 서비스의 느슨한 결합은 단순한 기술적 영향 이상의 영향을 미칩니다. 또한 소프트웨어 개발의 사회 기술적 측면에도 영향을 미칩니다. 결합은 '조직의 설계 시스템은 자체 커뮤니케이션 구조를 반영한다'는 개념인 콘웨이법의 근간입니다. 더 느슨하게 결합된 시스템은 더 분산되고, 확장 가능한 개발 접근 방식을 사용하여 더 느슨하게 결합된 조직을 의미합니다.

놀라운 사실

올해의 연구에서는 느슨하게 결합된 아키텍처가 팀의 번아웃을 유발하는 원인이 될 수 있는 것으로 드러났습니다. 이는 지난 몇 년간의 연구 결과와 상반되는 놀라운 사실입니다. Google의 분석에 따르면 정보 흐름이 자유로운 안정적인 팀에서는 번아웃 수준이 낮습니다. 웨스트럼의 생성적 문화와 팀 안정성은 둘 다 느슨하게 결합된 아키텍처를 지원하고 번아웃을 줄여주므로 이것은 분명히 모순되는 결과입니다. 단정적으로 결론을 내리기 전에 더 많은 연구가 필요한 부분입니다.

동시에 통합된 보안 조직에서 보안 요구사항을 정의하고 제어하는 경우 팀이 소프트웨어를 다른 팀과 분리하기가 더 어려울 수 있습니다. 이는 애플리케이션에 대해 가장 많은 책임을 지는 팀에 보안 문제를 이전하는 이점을 잘 보여줍니다(참조: [공급망 보안이 중요한 이유](#)). 이 경우는 조직에서 보다 미묘한 결합 형태 중 하나이며 보안에 대한 데이터를 수집했지만 다른 중앙 집중화된 기능에서도 이러한 상황이 발생할 가능성이 있습니다. 팀이 보안 및 기타 자주 중앙 집중화되는 기능에 대해 자체적으로 결정하도록 허용하는 것은 느슨하게 결합된 아키텍처를 사용하여 조직에 제공할 수 있는 이점을 누리는 방향으로 나아가는 한 가지 방법입니다.





다니엘라 비알바

문화

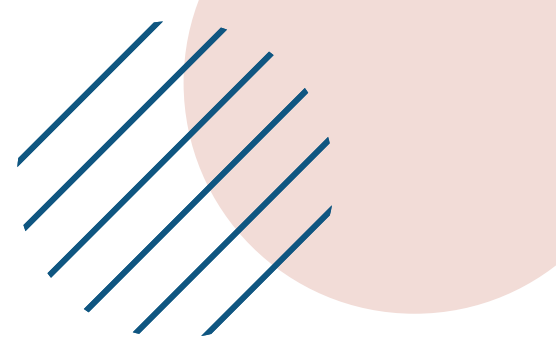
“이것이 여기서 일이 진행되는 방식입니다.”

다양한 업계에서 도전과 기회에 대한 조직의 접근 방식을 설명하는 데 이 문구를 수없이 사용했을 것입니다.

모든 조직에는 고유한 문화가 있으며, 연구 결과에 따르면 문화는 조직의 성공과 직원 복지의 기초가 되는 것으로 지속적으로 나타납니다.

가장 기본적인 수준에서 DevOps는 도구와 관행에 대한 것인 동시에, *소프트웨어를 빠르고 안정적이며 안전하게 개발하고 배포하기 위해 사람들이 협력하는 방법에* 관한 것이기 때문에 문화는 DevOps의 필수 요소이기도 합니다. 조직 문화에 영향을 미치는 요소를 이해하면 경영진이 문화 관련 문제를 정면으로 해결하는 데 도움이 될 수 있습니다. 따라서 조직은 건강한 문화를 조성하는 것에 우선순위를 두어야 합니다. 이러한 문화 관련 문제를 방치한다면 DevOps 관행이 정착되는 데 방해가 될 수 있습니다.





올해는 조직 문화의 건전성을 측정하기 위해 웨스트럼의 조직 유형을 계속 사용했습니다. 더 나아가 팀 이탈, 탄력 근무제, 체감되는 조직의 지원, 번아웃을 측정하여 문화에 대한 이해를 넓혔습니다.

올해의 연구 데이터는 **조직 성과**가 조직 내에 존재하는 문화 유형에 의해 영향을 받는다는 이전 연구 결과를 뒷받침합니다. 특히 **생성적 문화**는 관료적 또는 병리적 문화가 조성된 조직에 비해 더 높은 수준의 조직 성과로 이어집니다. 생성적 문화가

조성된 조직에서 일하는 직원은 안정적인 팀에 속할 가능성이 더 높고, 양질의 문서를 작성하며 대부분의 시간을 의미 있는 작업에 사용합니다.

팀 이탈

팀 이탈을 조사한 결과 **안정적인 팀**(지난 12개월 동안 팀 구성이 크게 변경되지 않은 팀)이 우수한 성과를 내는 조직에 있을 가능성이 더 높다는 것을 발견했습니다. 지속적인 이탈은 새로운 팀원이 온보딩하는 데 시간이 필요하기 때문에 생산성과 사기에 영향을 미칠 수 있습니다. 그리고 남아 있는 직원은 업무량과 팀 역학의 변화에 적응해야 할

웨스트럼 조직 문화

병리적 문화 권력 중심	관료적 문화 규칙 중심	생성적 문화 성과 중심
협력 수준 낮음	협력 수준 중간	협력 수준 높음
의사 전달 차단	의사 전달 무시	의사 전달 훈련
책임 회피	좁은 범위의 책임	책임 공유
연결 지양	연결 용인	연결 권장
실패가 책임 전가로 이어짐	실패가 처벌로 이어짐	실패가 질문으로 이어짐
새로운 아이디어가 억제됨	새로운 아이디어가 문제로 여겨짐	새로운 아이디어가 구현됨



수도 있습니다. 또한 연구에 따르면 안정적인 팀은 이탈이 많은 팀보다 양질의 문서를 작성한다고 보고할 가능성이 더 컸습니다. 지속적으로 변화에 적응해야 하는 팀은 양질의 문서 작성으로 이어지는 관행을 유지하는 데 어려움을 겪을 수 있습니다.

항상과 관련이 있는지 조사했습니다. 결과에 따르면 직원의 근무 방식에 있어 **탄력성**이 높은 조직이 보다 경직된 근무 방식을 가진 조직에 비해 조직 성과가 더 높은 것으로 나타났습니다. 이러한 결과는 직원에게 필요에 따라 업무 방식을 변경할 수 있는 자유를 부여하는 것이 조직에 실질적이고 직접적인 이점으로 이어진다는 증거를 제공합니다.

성과가 우수한 조직은 탄력 근무제를 시행할 가능성이 더 큼니다.

번아웃

번아웃은 업무와 관련해서 느끼는 두려움, 무관심, 냉소입니다. 번아웃을 경험하면 의욕이 없고 지칠 뿐만 아니라 직업 만족도가 낮아질 가능성이 높으며 이는 곧 높은 직원 이직률로 이어질 수 있습니다. 번아웃은 우울증과 불안, 심장 질환, 자살 생각의 위험 증가와 같은 다양한 정신적, 신체적 건강 결과와 관련이 있습니다.

탄력 근무제

코로나19 팬데믹 발병 이후 많은 조직이 탄력 근무제로 전환한 것을 감안하여 직원에게 원격, 대면 또는 하이브리드 근무 옵션 중에서 선택할 수 있는 자유를 주는 것이 조직 성과

지난해는 코로나19 팬데믹 상황에서의 번아웃을 측정했고 생성적 문화와 낮은 직원 번아웃 비율과의 연관성을 확인했습니다.

¹ Maslach C, Leiter MP. Understanding the burnout experience: recent research and its implications for psychiatry. World Psychiatry. 2016 Jun;15(2):103-11. doi: 10.1002/wps.20311. PMID: 27265691; PMCID: PMC4911781.

탄력 근무 모델은 직원의 번아웃 감소와 관련이 있으며 직원이 일하기 좋은 환경으로 소속 팀을 추천할 가능성을 높입니다.

올해도 이러한 결과를 똑같이 얻었으며 안정적인 팀과 탄력적인 근무 방식이 번아웃 감소와 관련이 있음을 보여줌으로써 번아웃에 대한 이해를 넓혔습니다. 또한 올해는 직원들이 자신의 팀을 친구나 동료에게 추천할지 여부를 나타내는 팀 추천 점수(NPS)를 측정했습니다. 그 결과 팀 NPS가 체감되는 경영진 지원과 관련이 있음을 확인했습니다. 그리고 번아웃 연구 결과를 반영하여, 생성적 문화, 안정적인 팀, 탄력 근무제가 직원들이 다른 사람에게 소속 팀을 추천할 가능성이 더 큰 것과 관련이 있음을 확인했습니다.

조직에 대한 직원의 인식

마지막으로, 직원들에게 향후 12개월 동안 소속 팀이 얼마나 많은 지원을 받을 것으로 예상하는지 질문하여 체감되는 경영진 지원을 조사했습니다. 결과는 체감되는 경영진 지원(예: 더 많은 재정 지원, 더 많은 리소스 할당, 후원) 수준이 높을수록 우수한 성과를 내는 조직과 관련이 있는 것으로 나타났습니다.

또한 직원들에게 향후 12개월 동안 보안 침해 또는 완전한 서비스 중단이 발생할 가능성을 예상하는 질문을 했습니다. 그 결과, 성과가 우수한 조직에서 일하는 직원들이 중대한 오류가 발생할 것이라고 예상할 가능성이 적고 조직에 대해 더 긍정적인 전망을 하고 있는 것으로 나타났습니다. 마찬가지로 우수한 소프트웨어 및 배포 성과를 내는 조직에서 일하는 직원이 비즈니스 성과를 개선하기 위해 현재 관행을 변경해야 한다고 느낄 **가능성이 적다**는 것을 발견했습니다.

대표성에 대해 덧붙이는 말

연구 결과에 따르면 소수 집단에 속한 직원은 성과가 우수한 조직에 속해 있는지 여부와 관계없이 무계획적 업무에 더 많은 시간을 할애한다고 보고할 가능성이 더 컸습니다. 또한 소수 집단에 속한 직원이 소수 집단에 속하지 않은 직원에 비해 더 높은 수준의 번아웃을 보고한 것으로 나타났습니다. 팀장은 업무량 불균형의 위험을 인식하고 팀원 간에 업무가 공정하게 할당되도록 해야 합니다.

모두 종합해 보면 이러한 연구 결과는 조직 및 팀 수준 모두에서 직원을 위한 건강하고 포용적인 환경을 만드는 것이 중요함을 강조합니다.

이렇게 문화의 중요성을 계속 강조하기는 하지만 조직의 문화를 바꾸거나 개선하는 것이 쉬운 일이 아님을 잘 알고 있습니다. 조직은 앞서서 직원의 경험을 이해하고 DevOps 혁신 노력의 일환으로 문화 관련 문제를 해결하는 데 리소스를 투자하는 것이 좋습니다.



04

공급망 보안이 중요한 이유



존 스피드 마이어



토드 쿨레자

2020년 11월에는 소프트웨어 공급망 보안 위기가 다가오고 있다고 의심하는 기술 전문가가 비교적 적었습니다. [Open Source Security Foundation](#)은 오픈소스 소프트웨어의 보안 문제를 해결하기 위해 설립된 조직입니다. 오픈소스 소프트웨어의 보안 문제 해결을 위한 노력을 계속 기울이고 있고 [긍정적인 발전](#)도 있었으나 주요 언론에서 주목받을 만한 성과를 거두지는 못했습니다. 중대한 공격이었던 [SolarWinds](#)는 모든 것을 바꾸어놓았습니다. 공격자가 트로이 목마

소프트웨어 업데이트를 통해 수천 개의 주요 기업 및 정부 네트워크에 조용히 침투하면 상황이 빠르게 변합니다.

오늘날 소프트웨어 공급망 보안 주제는 시급한 것으로 널리 인식되고 있습니다. 가족 식사 자리에서는 아니더라도 회의실에서는 자주 언급되는 주제입니다.



수많은 이니셔티브가 추진되고 있으며 소프트웨어 업계의 많은 부분에서 자체 소프트웨어 공급망 보안 관행을 개혁하고 공유 오픈소스의 보안을 개선하기 위해 노력했습니다.

이번 장에서는 두 가지 이니셔티브를 중점적으로 살펴봅니다. [SLSA](#)(Supply Chain Levels for Software Artifacts, '살사'로 발음)와 NIST [SSDF](#)(Secure Software Development Framework)가 그것입니다. 각 이니셔티브는 공격자가 소프트웨어 프로덕션 프로세스를 조작하고 악성 소프트웨어 업데이트를 통해 네트워크 방어자를 우회할 수 없도록 일련의 방어 수단을 제공합니다.

하지만 SLSA 및 SSDF와 관련된 소프트웨어 공급망 보안 관행이 얼마나 널리 사용되고 있을까요? 도입을 촉진하는데 지원이 필요한 관행은 무엇이고 이미 널리 사용되고 있는 관행은 무엇일까요? 지금까지 이러한 질문에 대한 체계적인 답변은 없었습니다. 수백 명의 소프트웨어 전문가를 대상으로 공급망 보안과 관련된 관행의 사용에 대한 설문조사를 진행하여 이에 대한 몇 가지 이른 답변을 제공합니다. 특히 다음과 같은 네 가지 주요 결과가 눈에 띕니다.

01 이미 채택을 시작함: SLSA 및 SSDF에 포함된 소프트웨어 공급망 보안 관행은 이미 어느 정도 채택되었지만 개선의 여지가 있습니다.

02 더 건강한 문화를 갖췄다면 유리함: 조직 문화는 소프트웨어 개발 보안 관행의 주요 동인이며 신뢰도가 높고 '비난을 삼가는' 문화에서는 신뢰도가 낮은 조직 문화보다 SLSA 및 SSDF 관행을 확립할 가능성이 더 높습니다.

03 핵심 통합 지점이 있음: 소프트웨어 공급망 보안의 기술적 측면을 채택하는 것은 주로 많은 공급망 보안 관행을 위한 통합 플랫폼을 제공하는 CI/CD의 사용에 달려 있는 것으로 보입니다.

04 뜻밖의 이점 제공: 더 나은 보안 관행은 보안 위험 완화 외에도 번아웃 감소와 같은 추가 이점을 제공합니다.

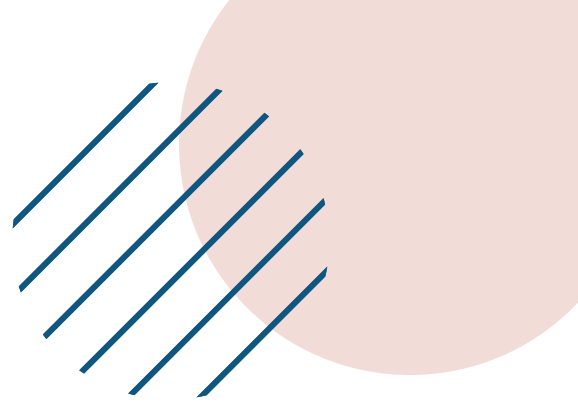
오늘날 기업이 보안 취약점을 방지하기 위해 취하는 조치

오늘날 조직이 구축한 소프트웨어의 보안 취약점을 식별하고 해결하기 위해 어떤 조치를 취하고 있는지 구체적으로 파악하기 위해 올해 설문조사에 24개 이상의 질문을 추가했습니다. 이러한 질문은 크게 두 가지 카테고리로 나뉩니다.

- 응답자에게 진술에 동의 또는 동의하지 않는지 여부를 묻는 질문(예: '조직에 보안 위협을 해결할 수 있는 효과적인 방법이 있습니다' 또는 '보안 테스트를 실행하는데 필요한 도구에 액세스할 수 있습니다').
- 응답자에게 조직의 보안 관행이 확립 또는 확립되어 있지 않은지 여부를 묻는 질문(예: '빌드는 빌드 스크립트를 통해서만 정의됩니다' 또는 '프로덕션 출시는 개발자의 워크스테이션이 아닌 중앙 집중식 CI/CD 시스템을 사용하여 빌드됩니다') 초기 테스트에서 응답자가 일부 보안 관련 질문에 동의하는 쪽으로 편중되었기 때문에 '확립/확립되지 않음' 척도를 사용했습니다. 하지만 SSDF에 대한 질문은 '동의/동의하지 않음' 응답 척도로 보다 자연스럽게 표현되었습니다.

SLSA 프레임워크(작성 당시 v0.1)는 SLSA '레벨'과 관련된 일련의 소프트웨어 공급망 무결성 관행을 명시하며 SLSA에서 높은 레벨은 높은 수준의 소프트웨어 공급망 보안을 보증하는 것을 의미합니다. 응답자들에게 SLSA와 관련된 여러 특정 관행에 대해 질문했습니다. 특히 '작업 중인 주요 애플리케이션 또는 서비스에 대해 다음 관행이 얼마나 확립되어 있습니까?'라는 질문으로 설문조사를 했습니다. 표 1에는 설문조사에서 다루는 SLSA 관련 관행에 대한 문구가 나열되어 있습니다.

SSDF(현재 v1.1)는 조직이 취약점이 적은 소프트웨어를 제공하도록 돕고 남아 있는 취약점의 잠재적 영향을 최소화하는 관행에 중점을 둡니다. SSDF 관행은 SLSA의 '레벨' 대신 조직 준비, 개발 중인 소프트웨어 보호, 안전한 소프트웨어 생성, 발견된 취약점에 대한 효과적인 대응이라는 4가지 카테고리로 그룹화됩니다. 설문조사를 통해 응답자에게 여러 SSDF 관행을 설명하는 진술에 얼마나 동의하는지(또는 동의하지 않는지) 물었습니다. 이러한 질문은 표 2에 요약되어 있습니다.



SLSA 관행	설문조사 정의
중앙 집중식 CI/CD	프로덕션 출시가 개발자의 워크스테이션이 아닌 중앙 집중식 CI/CD 시스템을 사용하여 빌드됩니다.
기록 보존	버전 및 변경 내역이 무기한 보존됩니다.
빌드 스크립트	빌드가 빌드 스크립트를 통해서만 완전히 정의됩니다.
격리됨	빌드가 격리됩니다. 동시 또는 후속 빌드를 방해할 수 없습니다.
빌드 텍스트 파일	빌드 정의 및 구성이 버전 제어 시스템에 저장된 텍스트 파일에 정의됩니다.
매개변수 메타데이터	아티팩트에 대한 빌드 메타데이터(예: 종속 항목, 빌드 프로세스, 빌드 환경)에 모든 빌드 매개변수가 포함됩니다.
종속 항목 메타데이터	아티팩트에 대한 빌드 메타데이터(예: 종속 항목, 빌드 프로세스, 빌드 환경)에 모든 종속 항목이 문서화됩니다.
메타데이터 생성됨	빌드 메타데이터(예: 종속 항목, 빌드 프로세스, 빌드 환경)가 빌드 서비스 또는 빌드 서비스를 읽는 빌드 메타데이터 생성기에 의해 생성됩니다.
입력 방지	빌드를 실행할 때 빌드 단계에서 빌드 입력을 동적으로 로드할 수 없습니다(예: 모든 필수 소스 및 종속 항목을 미리 가져옴).
사용자 수정 불가	빌드 서비스 사용자가 아티팩트에 대한 빌드 메타데이터(예: 종속 항목, 빌드 프로세스, 빌드 환경)를 수정할 수 없습니다.
메타데이터 제공	빌드 메타데이터(예: 종속 항목, 빌드 프로세스, 빌드 환경)가 필요한 사람에게 제공되며(예: 중앙 데이터베이스를 통해) 수용 가능한 형식으로 제공됩니다.
2인 검토	업데이트 기록에 있는 모든 변경사항이 제출 전에 신뢰할 수 있는 두 사람의 개별 검토 및 승인을 받아야 합니다.
메타데이터 서명됨	아티팩트가 생성된 방법에 대한 빌드 메타데이터(예: 종속 항목, 빌드 프로세스, 빌드 환경)가 내 빌드 서비스에 의해 서명됩니다.

표 1. SLSA 관련 설문조사 질문

참고: 응답자는 각 질문에 5가지(전혀 확립되지 않음, 약간 확립됨, 어느 정도 확립됨, 많이 확립됨, 완전히 확립됨)로 답변할 수 있었습니다.

SSDF 관행	설문조사 정의
보안 검토	내가 작업하는 애플리케이션의 모든 주요 기능에 대해 보안 검토가 수행됩니다.
지속적 코드 분석/테스트	지원되는 모든 출시에 대해 자동 또는 수동 코드 테스트 및 분석을 지속적으로 수행하여 이전에 감지되지 않은 취약점의 존재를 식별하거나 확인합니다.
초기 보안 테스트	본인 또는 다른 팀이 보안 테스트를 소프트웨어 개발 프로세스 초기에 실행합니다.
위험을 효과적으로 해결	조직에 보안 위험을 해결할 수 있는 효과적인 방법이 있습니다.
개발팀에 통합	보안 역할이 소프트웨어 개발팀에 통합됩니다.
요구사항 문서화	조직에서 개발하거나 획득한 소프트웨어(서드 파티 및 오픈소스 포함)에 대한 모든 보안 요구사항을 파악하고 문서화하는 프로세스가 마련되어 있습니다.
요구사항 정기적 검토	보안 요구사항을 정기적으로(매년 또는 필요한 경우 더 빨리) 검토합니다.
메타데이터 생성됨	빌드 메타데이터(예: 종속 항목, 빌드 프로세스, 빌드 환경)가 빌드 서비스 또는 빌드 서비스를 읽는 빌드 메타데이터 생성기에 의해 생성됩니다.
개발 주기에 통합	우리 회사에서는 소프트웨어 보안 프로토콜이 개발 프로세스에 원활하게 통합되어 있습니다.
프로젝트 전반에서 표준 프로세스	우리 회사에는 프로젝트 전반에서 소프트웨어 보안을 처리하는 표준 프로세스가 있습니다.
보안 보고서 모니터링	사용하는 소프트웨어 및 서드 파티 구성요소의 잠재적인 취약점과 관련하여 공개적으로 제공되는 정보를 모니터링하기 위해 지속적으로 노력하고 있습니다.
필요한 도구 있음	보안 테스트를 실행하는 데 필요한 도구에 액세스할 수 있습니다.

표 2. SSDF 관련 설문조사 질문

참고: 응답자는 각 질문에 7가지(전혀 동의하지 않음, 동의하지 않음, 약간 동의하지 않음, 동의하지도
동의하지 않지도 않음, 약간 동의함, 동의함, 매우 동의함)로 답변할 수 있었습니다.

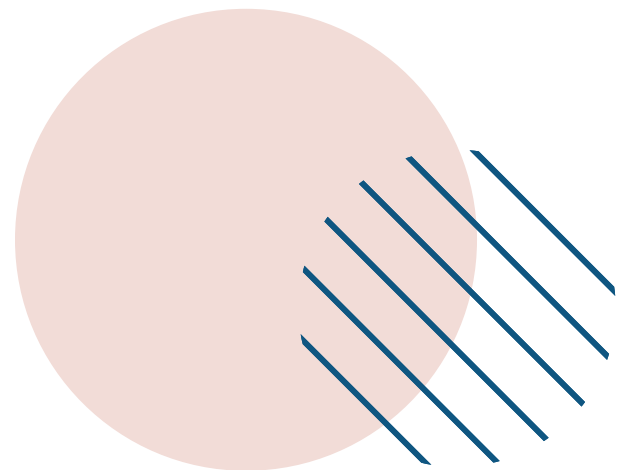


전반적으로 최신 업계 관행이 비교적 광범위하게 채택되고 있음을 확인했지만 이러한 관행이 더욱 확립될 여지는 충분합니다. 예를 들어 응답자의 66%는 '우리 회사에서는 소프트웨어 보안 프로토콜이 개발 프로세스에 원활하게 통합되어 있습니다'라는 진술에 동의했지만 매우 동의한다고 답한 응답자는 18%에 불과했습니다. 그림 1과 2는 보안 질문에 대한 참가자의 응답을 요약한 것입니다.

프로덕션 출시에 지속적 통합/지속적 배포(CI/CD) 시스템을 사용하는 것이 가장 일반적으로 확립된 관행임을 확인했으며 응답자의 63%는 '많이' 또는 '완전히' 확립되었다고 답했습니다. CI/CD가 이 목록에서 가장 높은 순위를 차지한 것은 [대부분의 조직이 CI/CD 프로세스의 일부로 애플리케이션 수준 보안 스캔을 구현한다고 밝힌 이전의 보안 연구 결과와 일치합니다.](#) 또한 별도로 진행된 보안 관련 정성적 인터뷰는 대부분의 개발자가 개발 중에 이러한 도구를 로컬에서 실행할 수 없음을 시사했습니다. 마찬가지로 SLSA 프레임워크 역시 공급망 보안을 위한 중앙 통합 지점으로 CI 시스템을 기반으로 합니다. 다음 섹션에서 설명하는 모델 분석에 따르면 조직에서 CI의 존재는 보안 관행의 성숙도를

예측하는 요인이었습니다. 따라서 이 중요한 인프라가 없다면 조직이 생성한 소프트웨어 아티팩트에 대해 일관된 스캐너, 린터, 테스트 세트를 실행하도록 보장하는 것은 매우 어려우리라 생각됩니다.

CI/CD 외에도 일반적으로 확립된 다른 관행에는 코드 기록의 무기한 보존(60%), 스크립트를 통해서만 정의되는 빌드(58%), 빌드를 서로 격리된 상태로 유지(57%), 소스 제어에 빌드 정의 저장(56%)이 포함됩니다. 가장 덜 일반적으로 확립된 2가지 관행은 두 명 이상의 검토자가 각 코드 변경을 승인하도록 요구하고(45%) 번조를 방지/감지하기 위해 빌드 메타데이터에 서명하는(41%) 것이었습니다.



확립된 관행에 대한 질문과 함께 참가자들에게 조직의 보안에 대한 일련의 진술에 동의하는지, 동의하지 않는지 질문했습니다. 동의 수준이 가장 높은 진술은 '사용하는 소프트웨어 및 서드 파티 구성요소의 잠재적인 취약점과 관련하여 공개적으로 제공되는 정보를 모니터링하기 위해 지속적으로 노력하고 있습니다.'로 응답자의 81%가 동의했습니다. 반대로 동의 비율이 가장 낮은 진술은 '소프트웨어 개발 시 보안 관행의 부정적인 영향에 대한 것이었는데, 응답자의 56%가 '회사에 존재하는 소프트웨어 보안 프로세스가 내가 작업하는 애플리케이션의 개발 프로세스를 더디게 만듭니다.'에 동의했습니다.

이 진술이 응답자 동의 수준이 가장 낮다는 점은 긍정적이지만, 대다수의 응답자가 현재 보안 프로세스가 개발 속도를 더디게 하고 있다고 답했다는 사실은 보안 도구 및 접근 방식을 개선할 여지가 많다는 것을 시사합니다. Google의 모델 분석도 이 해석을 뒷받침하며 소프트웨어 배포 성과에 대한 사소하지만 혼합된 영향을 보여줍니다.



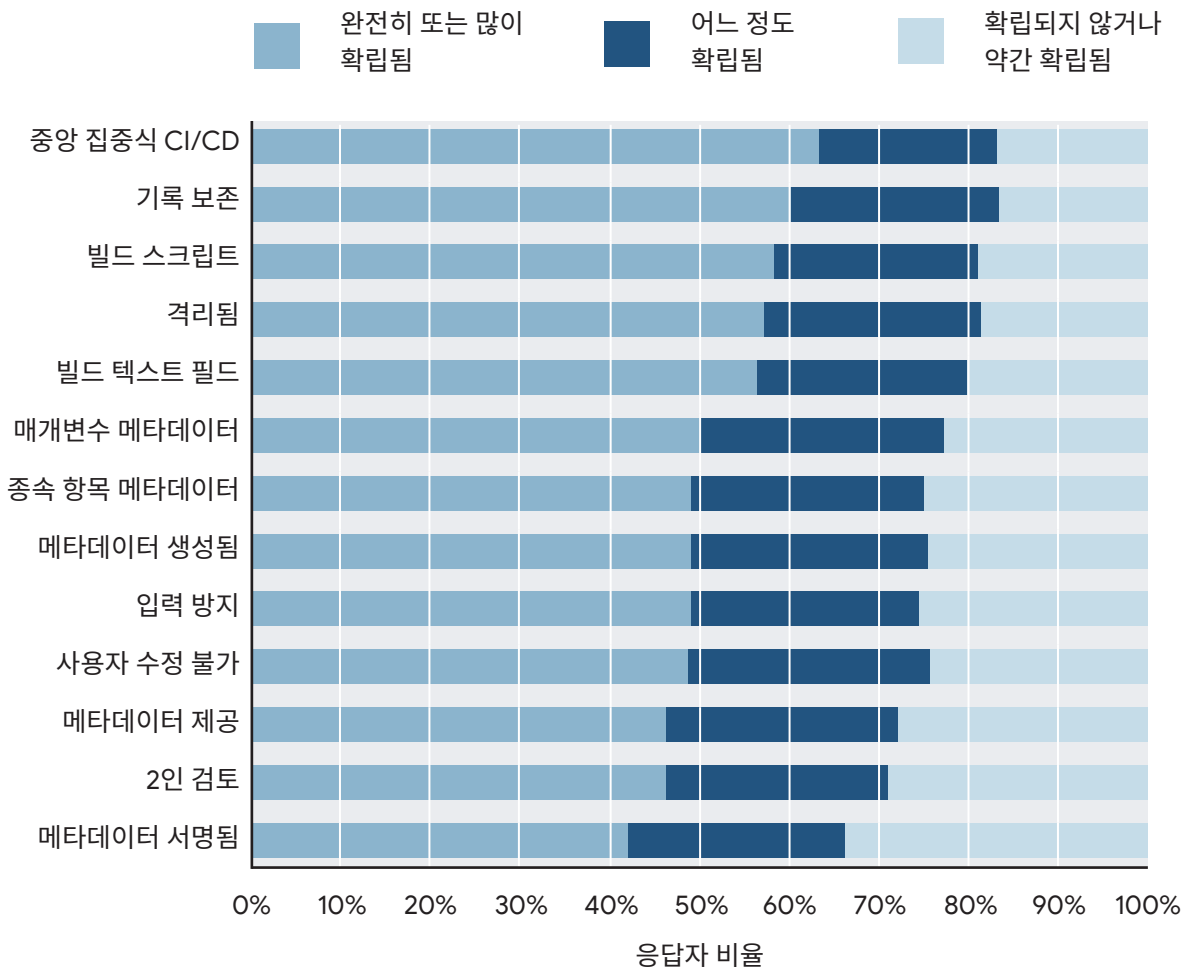


그림 1. SLSA 관행 확립

SLSA 관행 확립에 관한 설문조사 응답. 대다수의 응답자는 이러한 모든 관행이 어느 정도 확립되었다고 밝혔지만 아직까지는 상대적으로 소수만이 '완전히' 확립되었다고 답했습니다.

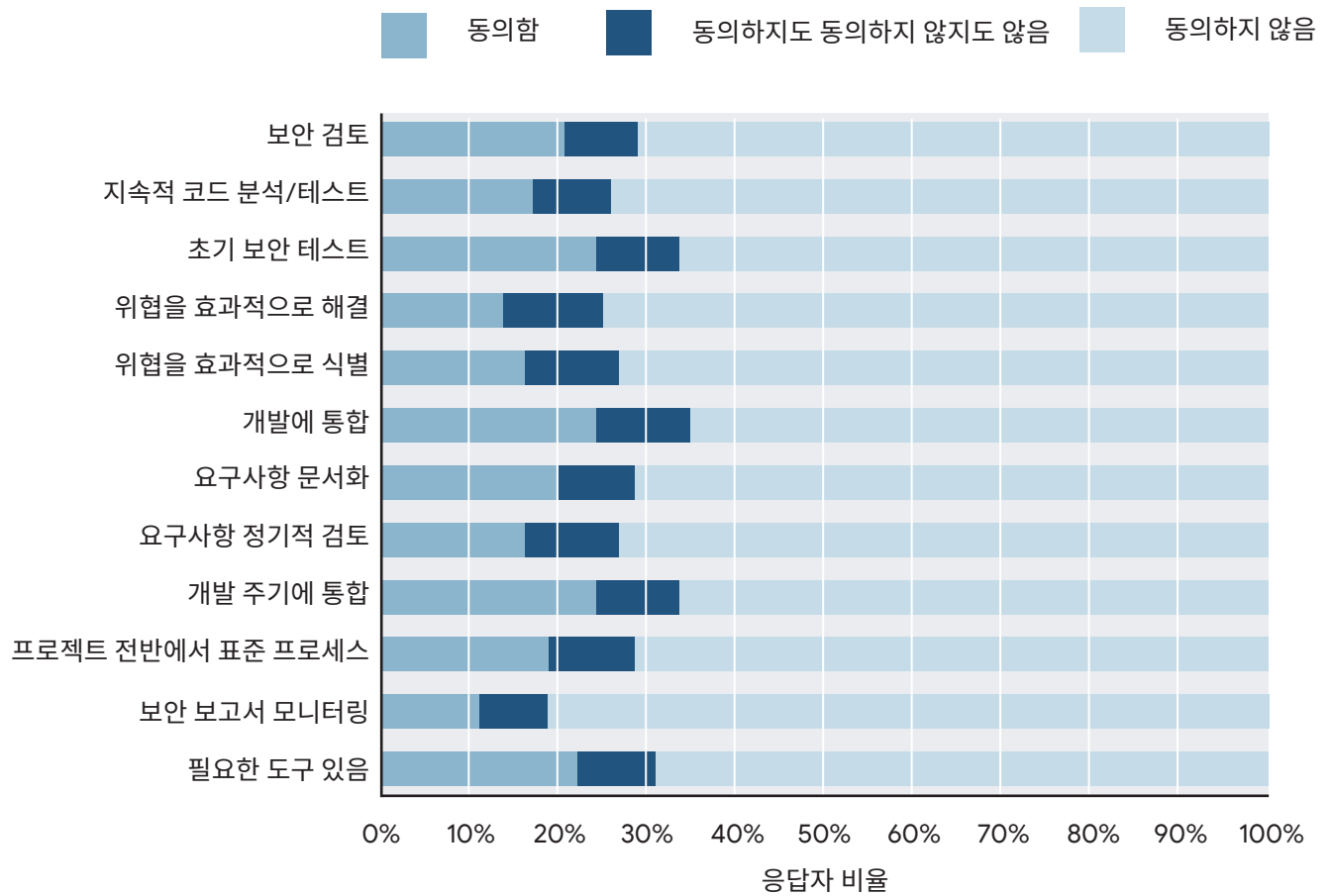


그림 2. SSDF 관행 확립

SSDF 관행 확립에 관한 설문조사 응답. SLSA와 유사하게 대다수의 응답자가 조직이 이러한 관행을 모두 따랐다는 데 동의했습니다.

기업이 우수한 보안 관행을 따르는 데 도움이 되는 것은 무엇일까요?

애플리케이션 보안은 소프트웨어 개발의 한 측면일 뿐이므로 개발자가 시간과 관심을 기울여야 하는 많은 요소 중 하나입니다. 큰 마찰을 일으키는 보안 접근 방식은 개발자의 불만을 야기할 뿐만 아니라 사람들은 마찰 요인을 피하려고 하기 때문에 전반적으로 비효율적일 수 있습니다. 예를 들어 전문 소프트웨어 엔지니어와의 일련의 연구 인터뷰에서 보안팀과의 터치 포인트가 프로젝트의 시작 또는 종료 시점으로 제한되어 있었고 팀이 소통하기 어려울 수 있다는 것을 확인했습니다. 한 참가자는 “애플리케이션 보안팀이 있지만 그들이 내 코드를 검토하도록 한 적은 없습니다. 대부분의 엔지니어들이 그렇듯이 보통 그들을 피합니다.” 라고 말합니다.

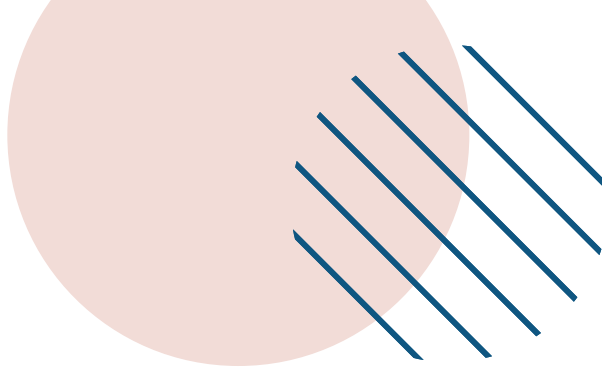
소프트웨어 보안을 개선하는 한 가지 접근 방법은 다음 보안 관행에 대한 장벽을 낮추는 것입니다. 대화에 응한 개발자는 옳은 일을 하기를 원했고 제공 기능이나 수정사항이 잠재적인 보안 문제보다 계속해서 우선시된다는 불만을 자주 언급했습니다. 예를 들어 별개의 보안 관련 설문조사를 진행한 한 응답자는 가장 큰 보안 문제가 “보안을 최우선 순위로 두는 것이며, 보안은 매력적인 분야도 아니고 제품을 더 많이 판매하는 것도 아니고 문제가 발생하기 전까지는 중요하지 않습니다.”라고 설명했습니다.

설문조사 데이터에 따르면 개발자가 ‘안전한 작업’을 더 쉽게 수행할 수 있도록 하는 여러 가지 요소가 있습니다.

설문조사를 통해 확인한 가장 큰 요소는 기술적인 요소가 전혀 아니었으며 문화적인 요소였습니다. **‘생성적’ 웨스트럼 문화 그룹에 가장 가까운 조직은 SLSA 프레임워크¹에 정의된 대로 광범위하게 확립된 보안 관행을 적용하고 있다고 말할 가능성이 훨씬 더 높았습니다.** 생성적 문화의 측면에는 긴밀하게 협력하고, 위험과 책임을 공유하며, 과거 실수로부터 배우는 문화가 포함됩니다. 우리는 이러한 문화적 특성이 소프트웨어 엔지니어가 공급망 보안에 대해 보다 능동적으로 행동하도록 장려하고, 직종에 관계없이 보안을 위해 노력한 사람에게 보상하거나, 잠재적인 보안 문제를 보고할 때 인식되는 위험을 줄이는 등의 다양한 방법으로 보다 건강한 보안 관행의 형태로 나타난다고 가정합니다.

기술적으로 보안을 주도하는 가장 중요한 3가지 요소는 인프라와 관련이 있습니다. 이는 직관적으로 타당한 표현입니다. 인프라가 취약점 스캔 또는 수동 코드 검토와 같은 작업을 보다 쉽게 수행할 수 있도록 한다면 엔지니어가 이를 사용할 가능성이 높아집니다. 특히, **소스 제어, 지속적 통합, 지속적 배포를 위한 시스템을 갖추는 것이 보다 확고하게**

¹ 흥미롭게도 이와 동일한 응답자가 NIST SSDF 질문에 동의할 가능성은 더 높지 않았습니다. SLSA와 SSDF가 애플리케이션 개발 보안의 서로 다른 측면을 논의하기는 하지만 이러한 질문 세트 간에 중복되는 부분이 있을 것으로 예상했습니다. 앞서 언급했듯이 SSDF의 응답 척도가 ‘동의’ 쪽으로 편향될 수 있는데 이러한 결과가 이 차이를 설명합니다.



확립된 SLSA 관행을 갖추는 것으로도 이어진다는 것을

확인했습니다. 핵심은 보안 문제가 개발자의 관심을 끄는 시점일 것이며 별개로 진행된 설문조사에서 주로 CI 기간 동안인 것으로 확인되었습니다. 일반적으로 CI는 코드 검토 바로 직전의 취약점 스캐너 및 기타 코드 분석 도구가 실행될 때 이루어집니다. 이렇게 하면 모든 코드 커밋이 동일한 보안 요구사항을 준수하도록 보장되기 때문입니다. 중앙 집중식 빌드 시스템이 없으면 이러한 일관된 스캔을 수행하기가 훨씬 더 어렵고, 소스 제어 기능이 없으면 중앙 집중식 빌드 시스템을 갖추기가 애초에 어렵습니다.

하지만 CI/CD의 일부인 보안 스캔은 소프트웨어 엔지니어에게 충분히 빠르지 않을 수 있습니다. 애플리케이션 개발자와 진행한 일련의 보안 관련 인터뷰에서 개발자는 개발 워크스테이션에서의 보안 스캔이 시간과 노력을 절약하는 데 도움이 될 것이라고 일관되게 말했습니다. 다음과 같은 두 가지 상황이 일반적으로 인용되었습니다. 1) 알려진 취약점이 있는 종속 항목을 기반으로 구축하고 있는지 미리 파악하여 이를 토대로 구축하기 전에 해당 종속 항목 사용을 재평가하려는 경우, 2) 현재 변경사항이 보안 문제를 해결했는지 여부를 확인하기 위해 때로는 몇 시간 단위로도 측정되는 긴 CI 대기 시간을 피하려는 경우. 두 경우 모두 소프트웨어 엔지니어는 CI '백스톱'이 필요하지만 동일한

보안 도구를 로컬에서 실행하는 기능이 더 빠르고 효율적으로 작업하는 데 도움이 될 것이라고 말했습니다.

위에서 논의한 문화적 및 기술적 요소는 보안의 가장 큰 동인이지만 유일한 요인은 아닙니다. 기타 주목할 만한 요소는 다음과 같습니다.

- 탄력적인 근무 제도(예: 조직에서 재택근무를 지원하는가?)
- 클라우드 사용(퍼블릭 또는 프라이빗)
- 클라우드 기반 애플리케이션 또는 서비스 작업
- 회사가 팀을 소중히 여기고 투자한다는 느낌
- 낮은 팀 이직률
- 조직 규모, 조직 규모가 클수록 더 높은 보안 점수 보고

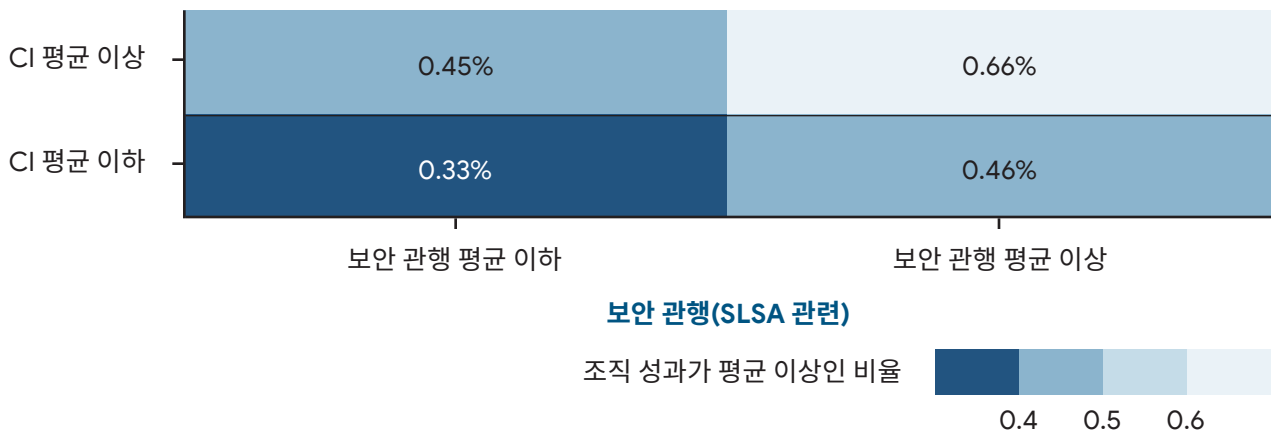
하지만 이러한 요소는 대체로 생성적 웨스트럼 문화(예: 탄력 근무제, 조직에서 인정받는 느낌, 낮은 팀 이직률) 또는 CI/CD 사용(예: 클라우드 기반 애플리케이션 작업 또는 대규모 조직에서 작업)과 관련이 있는 것으로 보입니다. 이 데이터는 조직 문화와 최신 개발 프로세스(예: 지속적 통합)가 조직의 애플리케이션 개발 보안의 가장 큰 동인이며 보안 태세를 강화하려는 조직이 시작하기에 가장 좋은 출발점이라고 믿게 합니다.

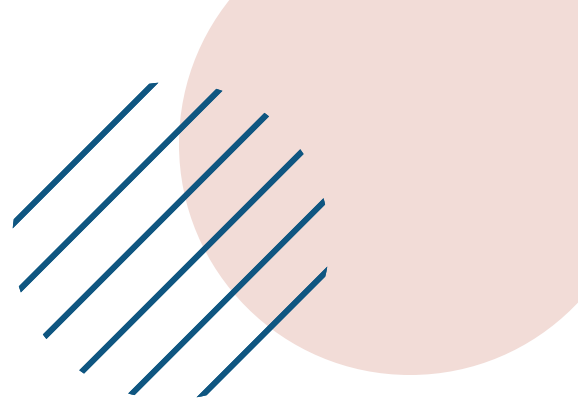
우수한 보안 관행은 어떤 결과를 가져오나요?

조직이 소프트웨어 개발과 관련된 보안 관행을 개선함에 따라 기대할 수 있는 이점은 무엇일까요? 설문조사 데이터에 따르면 참가자들은 **기업이 공급망 보안 관행을 더욱 확립함에 따라 보안 위반, 서비스 중단, 성능 저하가 발생할 가능성이 낮아질 것**으로 예상했습니다. 마찬가지로, 2022년 상반기에 별도로 진행한 연구에서 CI 중에 취약점 스캐너와 같은 도구를 실행하면 소프트웨어 종속 항목의 취약점을 식별할 가능성이 크게 증가한다는 사실을 발견했습니다. 이러한 도구를 사용한 응답자는 자체 코드 또는 종속 항목에서 보안 취약점을 식별했음을 보고할 가능성이 거의 두 배 높았습니다. 간단히 말해, SLSA와 SSDF 관행은 의도한 대로 작동하는 것으로 보입니다. 보안 위협을 없앨 수 있다고 단언할 수는 없지만

연구 결과는 조직의 보안 위험이 줄어든다는 것을 보여줍니다.

보안 관행은 성과 기반 결과에도 긍정적인 영향을 미칠 수 있지만 CI가 중추적인 역할을 한다는 의외의 사실을 유념해야 합니다. CI가 구현되지 않는다면 보안 관행은 소프트웨어 배포 성과에 영향을 미치지 않습니다. 그러나 CI가 구현된다면 보안 관행은 소프트웨어 배포 성과에 강력하고 긍정적인 영향을 미칩니다. 이는 근본적으로 보안 관행이 소프트웨어 배포 성과에 긍정적인 영향을 미치기 위해서는 CI가 필요하다는 것을 의미합니다. 또한 보안 관행은 일반적으로 조직 성과에 긍정적인 영향을 미치며 CI가 확고하게 확립되었을 때 그 효과는 증폭됩니다. 아래 그래프는 이러한 효과를 시각적으로 표현해본 것입니다.





응답자들은 인식되는 보안 위험 감소와 함께 팀원 간의 **번아웃 감소**를 보고했으며 자신의 조직을 **일하기 좋은 곳**으로 추천하려는 의지가 높아졌다고 보고했습니다. 이러한 두 연구 결과는 소프트웨어 엔지니어 입장에서 '좋은 점은 알지만 실천하기 어려운' 보안의 특성을 말해줍니다. 엔지니어 입장에서 이미 팍 찬 업무 일정에 업무가 하나 더 늘어나는 것입니다. 위협이 발견되었을 때 무계획적 업무나 '소방 훈련'과 달리 보안 관행을 기존 개발 워크플로에 통합하는 데 도움이 되는 도구 및 프로세스는 보안 위험을 완화하고 개발자 만족도를 높이는 메커니즘을 제공합니다.

종합하자면 연구 결과는 **건전하고 성과가 좋은 팀이 우수한 보안 관행**을 광범위하게 확립하는 경향이 있음을 시사합니다(앞서 언급한 바와 같이 개선의 여지는 계속 있음). SLSA 또는 SSDF 프레임워크와 같은 접근 방식을 따르는 것만으로는 우리가 측정하는 문화와 성과 측정항목을 모두 개선할 수는 없지만 보안을 위해 다른 개발 우선순위를 희생할 필요가 없다는 것은 분명합니다.

위협이 발견되었을 때 무계획적 업무나 '소방 훈련'과 달리 보안 관행을 기존 개발 워크플로에 통합하는 데 도움이 되는 도구 및 프로세스는 보안 위험을 완화하고 개발자 만족도를 높이는 메커니즘을 제공합니다.

05

놀라운 사실



데렉 드벨리스

매년 보고서는 해당 연도의 설문조사 응답에 중점을 두지만 State of DevOps Report 및 관련 주제 연구(예: 번아웃 및 문화에 대한 연구)의 전체 카탈로그 맥락에서 조사 결과를 이해하기 위해 최선을 다하고 있습니다. 재현 작업을 통해 이러한 효과의 신뢰성을 테스트하는 것이 이 연구 프로그램의 핵심 원칙입니다. 이를 통해 데이터에 맞게 신념을 조정하고 진화하거나 새로운 트렌드를 이해할 수 있습니다.

올해 우리는 몇 가지 놀라운 일을 경험했습니다. 이에 대한 잠재적인 이유는 무수히 많습니다. 우선, 올해 샘플에는 이전 보고서에 비해 경력이 적은 사람들이 더 많이 포함되었습니다. 이는 기술 관행의 구현을 감독하거나 지시할 책임이 있는 사람이 아니라 이러한 기술 관행 및 역량 실현을

직접 담당하는 사람의 의견을 더 많이 듣고 있다는 의미로 해석됩니다. 또 다른 가능성은 업계나 세계에서 무엇인가 변화가 있다는 것입니다. 어제 효과가 있었던 것이 내일도 효과적일 거라는 보장이 없습니다. 예를 들어 거시 경제적 영향과 코로나19 팬데믹의 그늘에 있었던 한 해는 DevOps의 작동 원리를 변화시켰을 수 있습니다. 마지막으로 모델에 포함된 항목의 미묘한 변화로 인해 변수 간의 관계가 변경되었을 수 있습니다.¹



¹ 주데아 펄의 'Book of Why'와 로버트 매셀리스의 'Statistical Rethinking'은 통계 모델에 포함하는 것과 포함하지 않는 것이 모델의 결과에 어떤 영향을 미칠 수 있는지에 대한 놀라운 예를 제공합니다.



예상치 못하거나 가설과 다른 결과로 인해 연구자는 보고서를 작성해야 하는 시점에 곤란한 상황에 놓이게 됩니다. 결과가 잘못되었거나 적어도 여러 연구의 경험적 증거를 뒷받침하지 않는(또는 심지어 모순되는) 위험한 상황을 감안할 때, 후속 연구를 진행하여 결과를 재현하고 해당 원인을 이해하려고 시도하는 것이 책임있는 행동입니다.² 연구원은 예상치 못한 결과에 집중한 나머지수년간의 연구를 통해 얼마나 많은 효과가 안정적으로 나타났는지를 간과할 위험도 있습니다. Google은 State of DevOps Report마다 100가지 이상의 과정에 대한 통계학적 연구를 진행합니다. 이렇게 함으로써 우연히 잘못된 결과를 얻을 수 있는 위험에 처합니다. 매년 연구를 재현함으로써 이에 대응하려고 노력합니다.

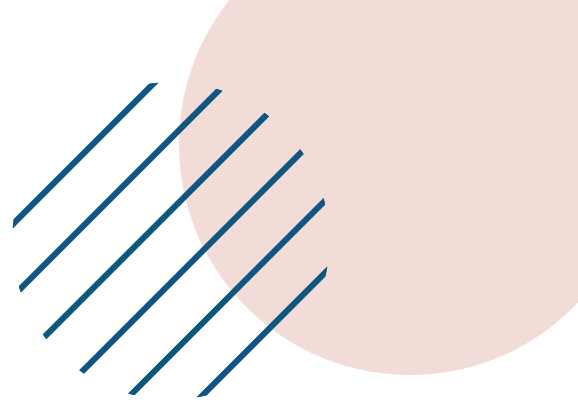
반면 이러한 결과를 보고하지 않으면 결과적으로 예상되거나 마음에 드는 내용은 잘 알려지고 예상치 못하거나 받아들이기 어려운 내용은 숨겨지는 서류함 효과³를 생성할 위험이 있습니다. Google은 균형을 유지하려고 노력합니다. 초기 연구 결과가 센세이션을 일으키는 것을 원하지는 않지만 이 내용을 공유하는 것 역시 중요하다고 생각합니다. 다음은 가장 놀라운 내용과 Google이 생각하는 의미입니다.

01 트링크 기반 개발 관행이 소프트웨어 배포 성과에 긍정적인 영향을 미친다는 것을 지속적으로 관찰해왔습니다. 실제로 이것은 2014년 이후 매년 연구에서 관찰되어왔습니다. 하지만 올해 트링크 기반 개발의 기능이 평소와는 다른 움직임을 보였습니다. 우선, 트링크 기능이 소프트웨어 배포 성과에 부정적인 영향을 미쳤습니다. 이전 연구 보고서에서는 그 반대였습니다. 이 연구 결과가 얼마나 변칙적인지를 감안할 때 향후 연구에서 이 현상이 재현되는지 확인하고 커뮤니티에서 설명할 수 있는지 듣고 싶습니다.

02 운영 성과 역시 우수할 때만 소프트웨어 배포 성과가 조직 성과에 도움이 된다는 사실을 발견했으며 많은 응답자의 운영 성과가 우수하지 않았습니다. 이것은 소프트웨어 배포 성과와 조직 성과 사이의 연관성이 훨씬 더 명확했던 이전의 반복된 연구와 모순됩니다.

² Kerr, N. L. (1998). HARKing: Hypothesizing after the results are known. *Personality and social psychology review*, 2(3), 196-217.

³ Rosenthal, R. (1979). The file drawer problem and tolerance for null results. *Psychological bulletin*, 86(3), 638.



03 문서화 관행은 소프트웨어 배포 성과에 부정적인 영향을 미쳤습니다. 이것은 이전 보고서와 상충됩니다. 한 가지 가설은 특히 성과가 우수한 팀 사이에서 문서화 작업이 점점 더 자동화되고 있다는 것입니다. 추가 데이터를 수집할 때까지는 이러한 믿음을 지지하거나 반박할 증거가 거의 없습니다.

04 일부 기술적 역량(예: 트렁크 기반 개발, 느슨하게 결합된 아키텍처, CI, CD)은 번아웃으로 이어지는 것처럼 보였습니다. 위에서 언급한 바와 같이, 이 샘플에서는 이전 연도 샘플의 참가자들보다 경력이 비교적 적은 응답자가 많았습니다. 따라서 이니셔티브를 만들거나 감독하는 책임이 있는 사람이 아니라 역량 실현을 담당하는 사람을 조사했을 수 있습니다. 구현 프로세스는 이 프로세스를 감독하는 것보다 훨씬 더 어려울 수 있습니다. 이 결과를 더 잘 이해하기 위해 추가 연구를 진행하고자 합니다.

05 안정성 엔지니어링 관행은 소프트웨어 배포 성과에 부정적인 영향을 미쳤습니다. 한 가지 설명은 그 인과 관계가 밀접하지 않을 수 있다는 것입니다. 올해 새로운 클러스터링 분석('비교 방법' 참조)에서는 클러스터의 하위 집합이 소프트웨어 배포 성과를 무시하면서 신뢰성에 초점을 맞추는 것으로 나타났습니다. 하나가 다른 하나 없이 실현될 수 있다는 점에서 두 가지가 별개라고 생각하지만 결국 소프트웨어 배포 성과가 조직 성과에 기여하려면 신뢰성이 갖추어져야 합니다.

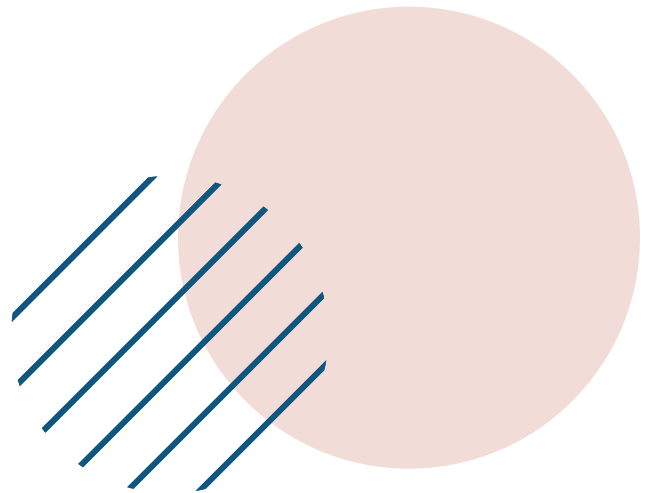
06 팀이 안전한 소프트웨어 공급망을 유지하기 위해 이러한 접근 방식을 채택하고 있는지 파악하기 위해 SLSA 관련 관행을 추가했습니다. 보안 관행의 구현과 성과(예: 기술적 역량 사용, 더 나은 소프트웨어 배포 성과, 더 나은 조직 성과) 사이에 어느 정도 연관성이 있을 것으로 예상했지만 보안 관행이 실제로 기술적 역량이 소프트웨어 배포 성과와 조직 성과에 영향을 미치는 메커니즘이라는 사실에 놀랐습니다.



SLSA 관련 관행을 포함함으로써 지속적 통합, 버전 제어, 지속적 배포가 소프트웨어 배포 성과와 조직 성과 둘 다에 미치는 영향 대부분이 설명되는 것으로 보입니다. 달리 말하면, 많은 기술적 역량이 SLSA 관련 관행에 긍정적인 영향을 미치고, SLSA 관련 관행에 대한 이러한 긍정적인 영향을 통해 소프트웨어 배포 성과와 조직 성과 둘 다에 긍정적인 영향을 미치는 인과관계가 데이터에서 감지되었습니다. 이 결과를 감지하는 데는 매개 분석을 사용했습니다.⁴⁵ 따라서 SLSA 관련 관행에 대한 척도가 팀의 다른 기능(예: 일반 성과)을 추적하는지 여부와 보안 관행이 더 나은 소프트웨어 배포 성과 및 조직 성과로 이어지는 방식을 조사해야 합니다.

내년에 이러한 효과를 다시 연구하여 이러한 새로운 패턴을 재현하고 설명할 수 있는지, 아니면 이러한 패턴을 이상점으로 무시해야 하는지(이 또한 설명이 필요함)를 확인할 수 있기를 바랍니다. 언제나 그렇듯이, 커뮤니티의 피드백을 환영합니다.

[DORA Community\(http://dora.community\)](http://dora.community)에 가입하여 올해 보고서의 이러한 놀라운 사실과 기타 연구 결과에 대한 토론을 계속하세요.



⁴ Jung, Sun Jae. "Introduction to Mediation Analysis and Examples of Its Application to Real-world Data." Journal of preventive medicine and public health = Yebang Uihakhoe chi vol. 54,3 (2021): 166-172. doi:10.3961/jpmph.21.069

⁵ Carrión, Gabriel Cepeda, Christian Nitzl, and José L. Roldán. "Mediation analyses in partial least squares structural equation modeling: Guidelines and empirical examples." Partial least squares path modeling. Springer, Cham, 2017. 173-195.

06

인구통계 및 기업통계



데렉 드벨리스

이번 연구에 참여하시고 업계에 기여해 주셔서 감사합니다.

설문조사 대상

State of DevOps Report는 8년간의 연구와 업계 전문가의 33,000건이 넘는 설문조사 응답을 통해 팀과 조직을 가장 성공적으로 이끄는 소프트웨어 개발 및 DevOps 관행을 제시합니다. 올해는 전 세계의 다양한 업계에서 일하는 1,350 명이 넘는 전문가들이 경험을 공유하여 성과를 향상하는 요인에 대한 이해를 높이는 데 도움을 주었습니다. 이번 연구에 참여하시고 업계에 기여해 주셔서 감사합니다. 요약하면, 인구통계 및 기업통계 전반에서 대표성이 매우 일관적으로 유지되었습니다.

예년과 마찬가지로 각 설문조사 응답자로부터 인구통계 정보를 수집했습니다. 카테고리에는 성별, 장애인, 소수 집단이 포함됩니다.

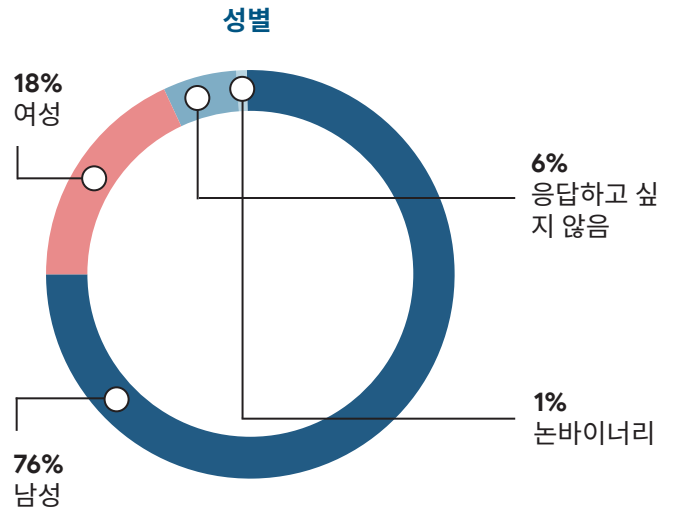
올해는 회사 규모, 업종, 지역을 포함한 기업통계학적 카테고리 전반에 걸쳐 대표성이 이전 보고서와 일관되게 유지되었습니다. 이번에도 60%가 넘는 응답자가 엔지니어 또는 관리자에 해당하며 3분의 1은 기술 업계에서 근무합니다. 또한 응답자에는 금융 서비스, 소매, 산업/제조회사 종사자도 포함되었습니다.



인구통계

성별

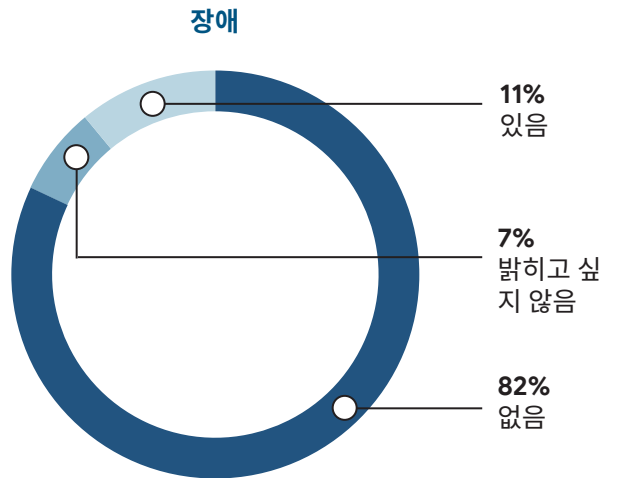
2021년에 비해 올해 샘플은 여성 응답자의 비율이 더 높았습니다(18% 대 12%). 남성 응답자의 비율(76%)은 2021년(83%)보다 낮았습니다. 응답자들은 팀원의 25%를 여성이 차지한다고 밝혔으며 이는 2021년(25%)과 동일합니다.



여성 비율: 25% 증양값

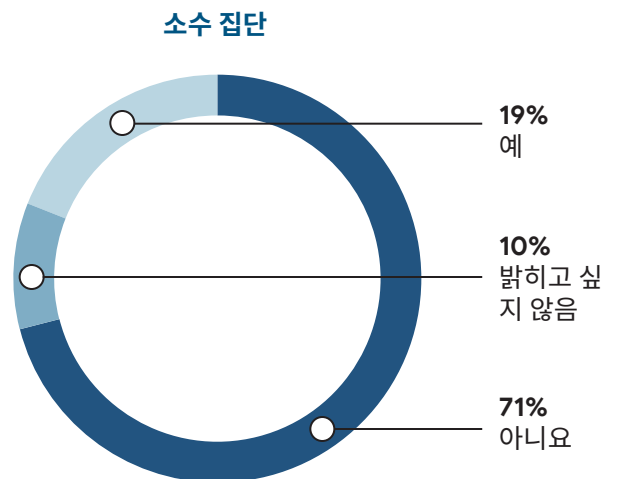
장애

장애는 [Washington Group Short Set](#)의 안내를 따르는 6가지 기준으로 식별했습니다. 장애에 대한 질문을 한 것은 올해가 4년째입니다. 장애인 비율은 2021년 보고서와 마찬가지로 11%였습니다.



소수 집단

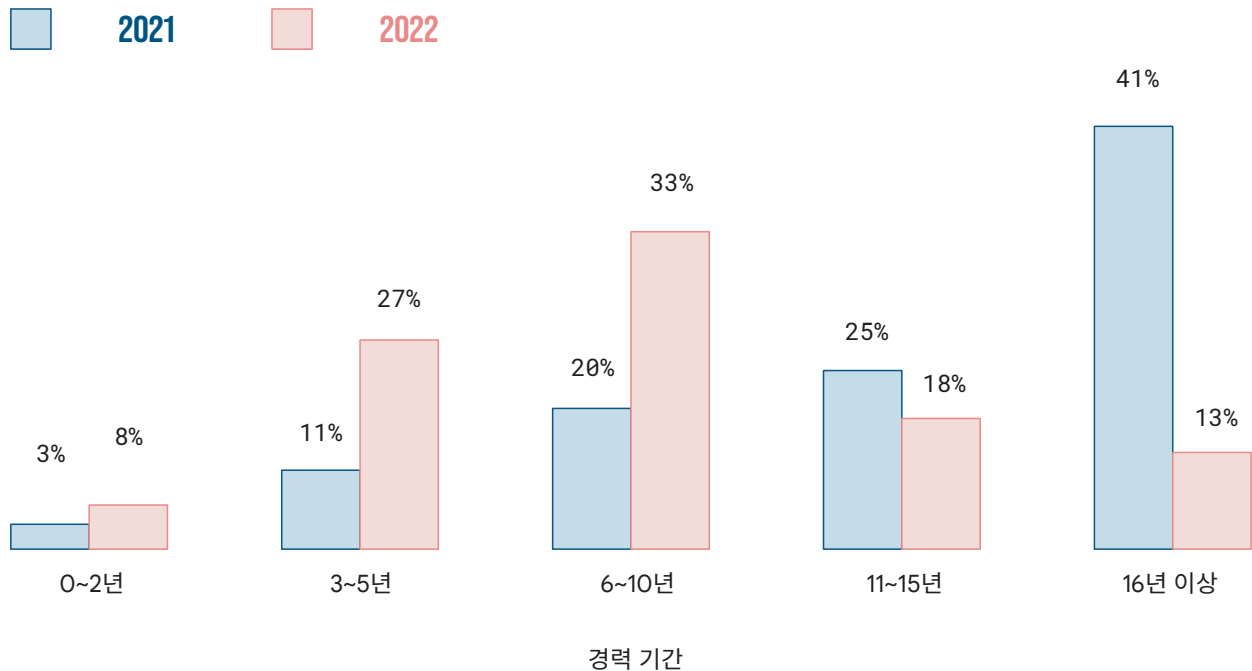
소수 집단의 구성원으로 식별하는 것은 인종, 성별 또는 다른 특성을 나타낼 수 있습니다. 소수 집단에 대한 질문을 한 것은 올해가 5년째입니다. 소수 집단에 속하는 것으로 식별되는 응답자의 비율은 2021년 17%에서 2022년 19%로 약간 증가했습니다.

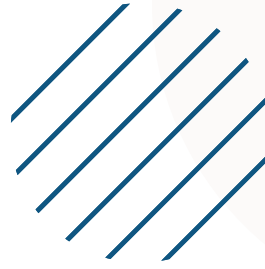


경력 기간

특히 올해는 2021년(14%)보다 경력이 5년 이하인 응답자 (35%)가 더 많았습니다. 따라서 당연히 16년 이상의 경력을 가진 응답자 비율(13%)은 2021년(41%)의 극히 일부에 불과했습니다.

이러한 변화로 데이터에 나타난 몇 가지 패턴을 설명할 수 있으며 결과를 해석할 때, 특히 지난해와 비교할 때 이를 염두에 두는 것이 중요하다고 생각합니다.

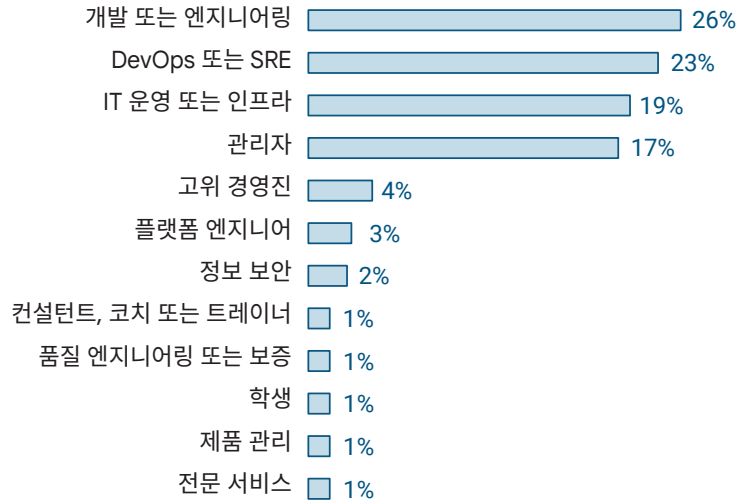




기업통계

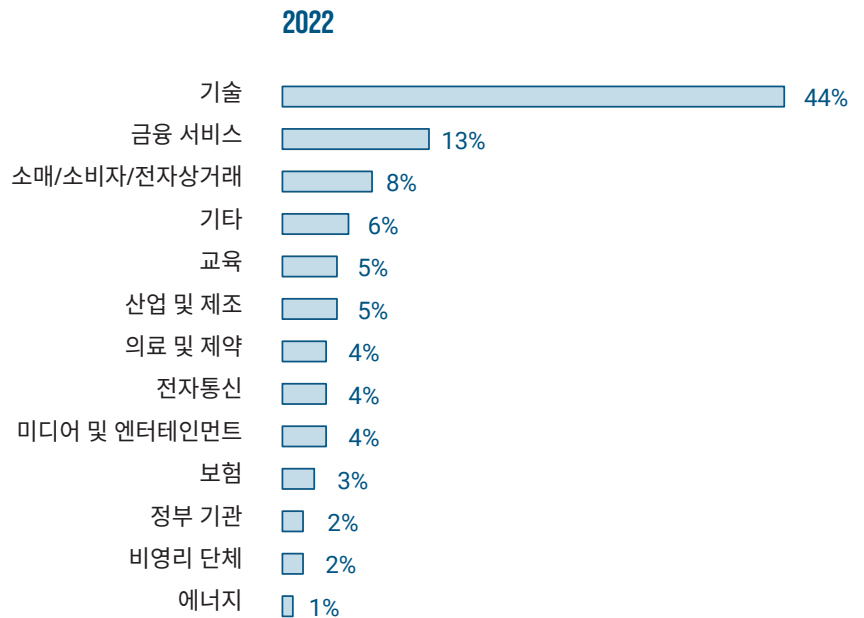
역할

응답자의 85%는 개발 또는 엔지니어링팀 (26%), DevOps 또는 SRE팀(23%), IT 운영 또는 인프라팀(19%)에서 근무하는 개인이거나 관리자(17%)로 구성됩니다. IT 운영 또는 인프라팀에서 근무하는 응답자의 비율(19%)은 지난해 비율(9%)에서 두 배 이상 증가했습니다. 지난해에 비해 크게 감소한 두 그룹은 고위 경영진(2021년 9%에서 4%)과 전문 서비스(2021년 4%에서 1%)입니다.



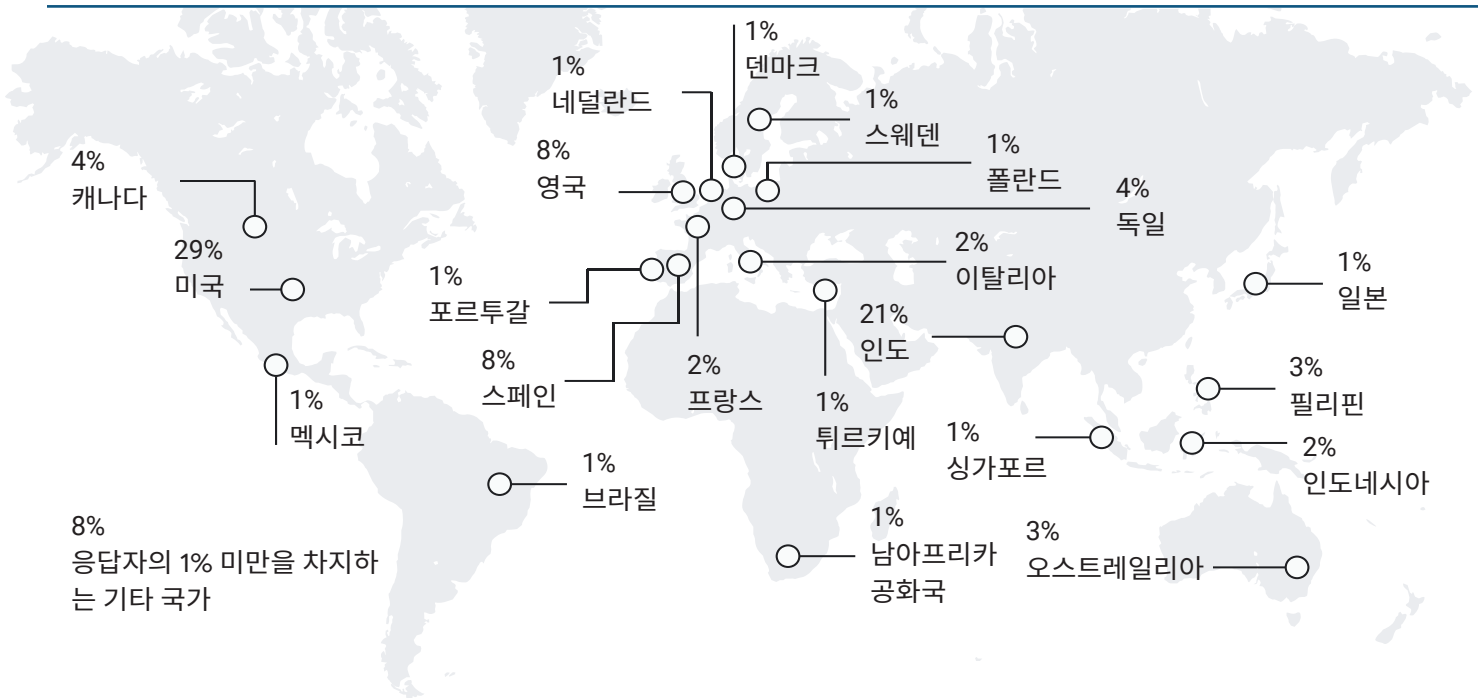
업종

이전의 State of DevOps Report에서와 같이 대부분의 응답자가 기술 업계 종사자이며 금융 서비스업, 기타 업종, 소매업이 그 뒤를 잇고 있습니다.



지역

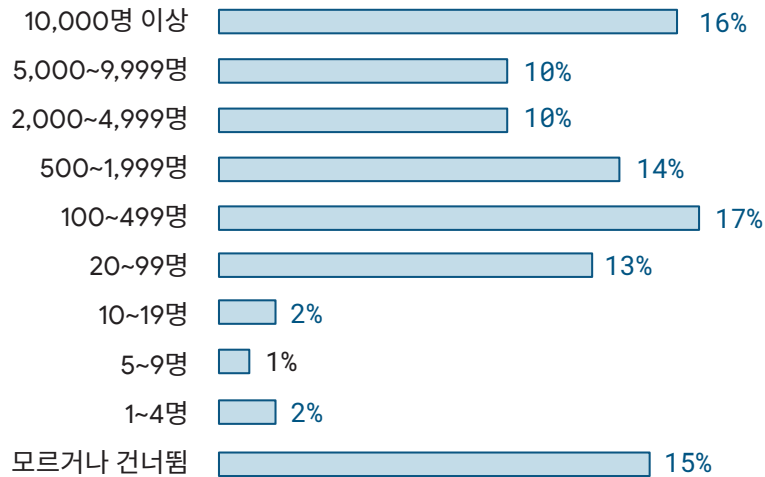
올해는 응답자들에게 지역 대신 출신 국가를 선택해달라고 요청했습니다. 대륙으로 자주 표현되는 지역은 응답자의 구성을 이해하기에는 너무 일반적인 것 같습니다. 70개국 이상의 참가자들로부터 응답을 받았으며 응답자의 89%는 22개국 출신입니다.

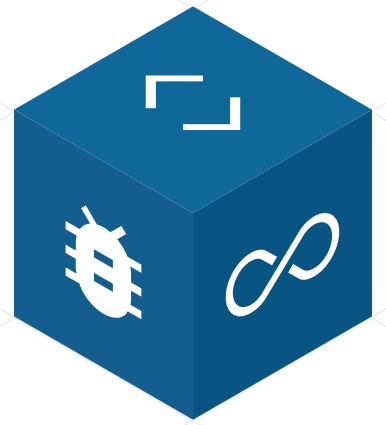


직원 수

이전 State of DevOps 설문조사와 일관되게 응답자는 다양한 규모의 조직에 몸담고 있습니다. 응답자의 22%는 직원이 10,000명 이상인 회사 소속이며 7%는 직원이 5,000~9,999명인 회사 소속입니다. 응답자의 다른 15%는

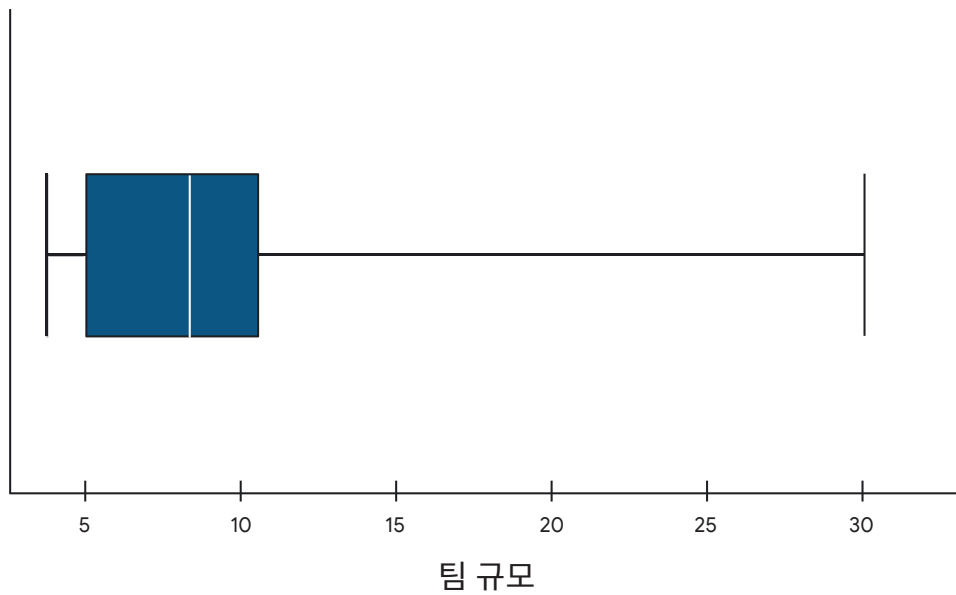
직원이 2,000~4,999명인 조직에 있습니다. 또한 직원이 500~1,999명(13%), 100~499명(15%), 마지막으로 20~99명(15%)인 조직에 속한 응답자의 비율도 비슷했습니다. 또한 올해는 응답자들이 조직의 규모에 대해 '모름'을 선택할 수 있도록 허용했습니다. 응답자의 15%는 모른다고 답했거나 질문을 건너뛰었습니다.





팀 규모

올해는 참가자들에게 대략적인 팀원 수를 표시하도록 했습니다. 응답자의 25%는 5명 이하로 구성된 팀에서 일했습니다. 응답자의 50%는 8명 이하로 구성된 팀에서 일했습니다. 응답자의 75%는 12명 이하로 구성된 팀에서 일했습니다.

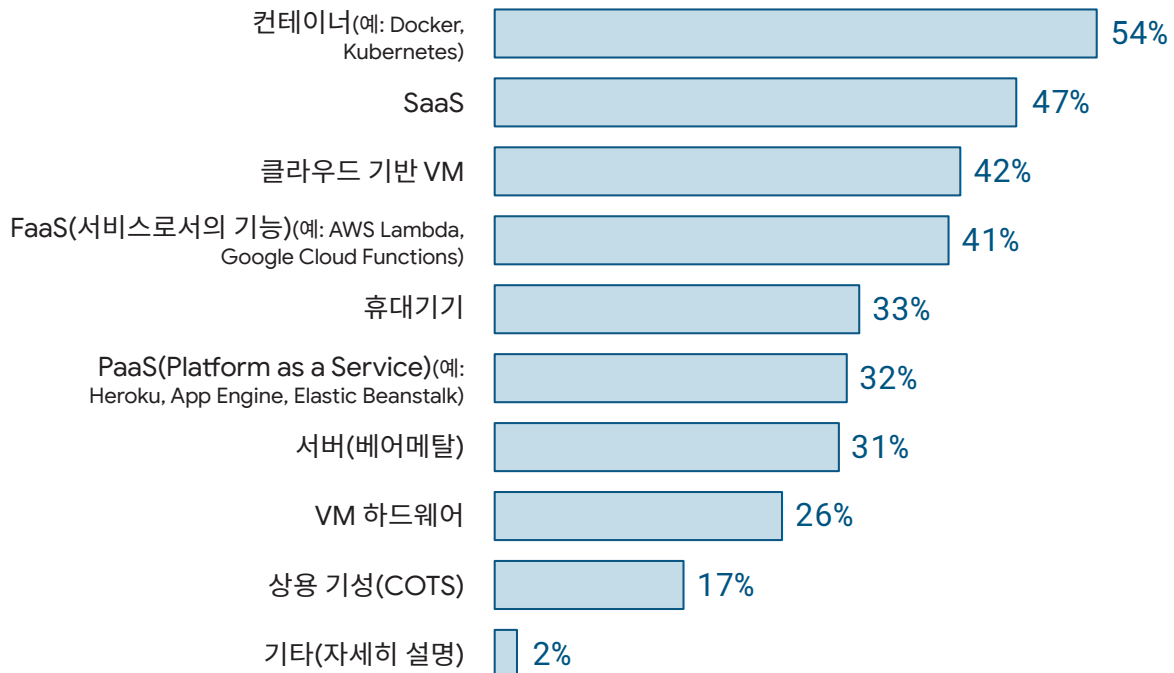


배포 대상

2021년은 응답자가 작업 중인 주요 서비스 또는 애플리케이션을 어디에서 배포했는지 살펴보기로 결정한 첫 해였습니다. 놀랍게도 가장 많은 배포 대상은 컨테이너였습니다. 올해도

다르지 않았지만 지난해(64%)보다 낮은 비율(54%)이었습니다.

또한 응답자들이 배포 대상을 반영할 수 있는 더 많은 방법을 제공하기 위해 더 많은 옵션을 추가했습니다.



07

최종 의견



데렉 드벨리스

해마다 이 보고서를 작성하면서 관행과 역량이 조직 성과와 같이 비즈니스에 중요한 결과에 어떻게 영향을 미치는지 철저히 분석하기 위해 노력합니다. 이전 보고서에 발표된 많은 효과의 재현성을 평가하고 DevOps 영역에서 새롭게 등장하는 최우선 과제를 설명하기 위해 연구 범위를 확장해나갑니다. 올해는 보안 관행을 심층적으로 살펴보기 위해 설문조사와 분석을 구상했으며, 특정 효과의 조건부 또는 종속성을 조사하기 위해 통계 모델링 접근 방식도 수정했습니다. 소프트웨어 배포 및 운영 성과 환경을 설명할 새로운 방법도 모색했습니다.

여러 면에서 올해 구체적으로 논의된 내용은 이전의 여러 보고서에서 다루었던 내용을 상기시키는 것입니다. 기술적 역량은 상호 보완적으로 사용될 때 더 우수한 성과로 이어지고, 클라우드 사용에는 내재된 많은 이점이 있으며, 직장 문화와 탄력적인 근무 형태는 조직의 성과 향상으로 이어집니다. 또한 직원 번아웃은 조직이 목표를 달성하는 것을 방해합니다. 모델에 명시적으로 추가한 상호작용 분석은 특정 효과가 발생할 수 있는 조건을 이해하는 데 도움이

되었습니다. 예를 들어 소프트웨어 배포 성과는 운영 성과 (신뢰성)가 우수할 때만 조직 성과에 긍정적인 영향을 미치는 것으로 보이며, 따라서 조직의 성장을 위해서는 둘 다 필요하다는 결론으로 이어집니다. 또한 관련 섹션에서 강조한 몇 가지 놀라운 사실도 있었습니다.

올해 설문조사에 참여해 주신 모든 분들께 감사드리며, 이 연구가 설문 참여자와 해당 조직이 더 나은 팀과 소프트웨어를 구축하는 데 도움이 되면서도 일과 생활의 균형을 유지하는 데 보탬이 되기를 바랍니다.



08

감사의 말씀

많은 분들의 적극적인 참여로 올해 보고서를 완성할 수 있었습니다. 설문조사 질문 설계, 분석, 작성, 편집, 보고서 디자인은 이 거대한 노력을 실현할 수 있도록 동료들이 도움을 준 많은 방법 중 일부일 뿐입니다. 올해 보고서에 대한 의견과 안내를 제공해주신 모든 분들께 감사 말씀을 드립니다. 감사 인사를 전할 모든 분들의 이름은 영문 이름의 성을 기준으로 알파벳순으로 나열되어 있습니다.

스캇 어코인

알렉스 배럿

제임스 브룩뱅크

김 카스티요

롤리 체시

젠나 데일리

데렉 드벨리스

롭 에드워즈

데이브 팔리

크리스토퍼 그랜트

마슈하드 해리

네이슨 하비

다미스 카루나라트네

토드 쿨레자

아만다 루이스

이안 루이스

에릭 맥스웰

존 스피드 마이어

스티브 맥기

저신다 메인

엘리슨 밀리건

파블로 페레스

빌라누에바

클레어 피터스

코너 포스키

데이브 스탠크

더스틴 스미스

세스 바고

다니엘라 비알바

브레나 워싱턴

카이위안 '프랭크' 수

니콜라 얍

09

저자



클레어 피터스

클레어 피터스는 Google의 사용자 경험 연구원입니다. 응용 환경에서 DORA의 다양한 측면과 표현 등을 연구합니다. 클레어는 팀과 개인이 일상 업무에서 DORA 원칙을 보다 효과적으로 적용할 수 있도록 돕는 것을 목표로 Google 클라우드 애플리케이션 현대화 프로그램(CAMP), DORA 기반 고객 참여, Four-Keys 관련 도구를 연구합니다. 또한 DORA의 핵심 리서치팀의 멤버이며 연간 DORA 설문조사 및 State of DevOps Report 작성에 참여합니다. 코펜하겐 대학교에서 응용 문화 분석 석사 학위를 받았습니다.



데이브 팔리

데이브 팔리는 Continuous Delivery Ltd의 상무이사 겸 설립자이며 Continuous Delivery YouTube 채널의 크리에이터입니다. 베스트셀러인 Continuous Delivery 도서의 공동 저자입니다. 역시 베스트셀러인 Modern Software Engineering: Doing What Works to Build Better Software Faster의 저자이기도 합니다. Reactive Manifesto의 공동 저자이며 오픈소스 LMAX Disruptor 프로젝트로 Duke Award를 수상했습니다. 데이브는 지속적 배포의 선구자이고, CD, DevOps, TDD, 소프트웨어 설계 분야의 사고 리더이자 전문가이며 성과가 우수한 팀을 만들고, 성공을 위한 조직을 개편하고, 뛰어난 소프트웨어를 만드는 데 많은 경력을 가지고 있습니다. 데이브에 대해 자세히 알고 싶으시면 그의 [Twitter](#), [YouTube](#), [블로그](#), [웹사이트](#)를 방문하세요.



다니엘라 비알바

다니엘라 비알바는 DORA 프로젝트 전담 사용자 경험 연구원입니다. 다니엘라는 개발자에게 만족감과 생산성을 선사하는 요소를 파악하는 연구에 주력합니다. Google에 합류하기 전에 다니엘라는 명상 훈련의 이점, 대학생의 경험에 영향을 미치는 심리 사회적 요인, 목격자 기억, 거짓 자백을 연구했습니다. 다니엘라는 플로리다 인터내셔널 대학교에서 실험 심리학 박사 학위를 받았습니다.



데이브 스탠크

데이브 스탠크는 Google의 개발자 관계 엔지니어로, 고객에게 DevOps 및 SRE 채택을 위한 권장사항에 대해 조언합니다. 경력 전반에 걸쳐 스타트업 CTO, 제품 관리자, 고객 지원, 소프트웨어 개발자, 시스템 관리자, 그래픽 디자이너를 비롯한 다양한 역할을 수행했습니다. 데이브는 컬럼비아 대학교에서 기술 관리 석사 학위를 받았습니다.



데렉 드벨리스

데렉 드벨리스는 Google의 정량적 사용자 경험 연구원입니다. Google에서 데렉은 설문조사 연구, 로그 분석, 제품 개발의 핵심 개념을 측정하는 방법을 파악하는 일을 주로 합니다. 최근에는 인간 대 AI 상호작용, 코로나19 발병이 금연에 미치는 영향, NLP 오류에 대한 설계, 개인 정보 보호 논의에서 UX의 역할을 발표했습니다.



에릭 맥스웰

에릭 맥스웰은 Google에서 DevOps 디지털 혁신 관행을 이끌고 있으며 세계 최고 기업을 대상으로 조금씩, 지속적으로 개선하는 방법에 대해 조언합니다. 에릭은 엔지니어로서 최전선에서 모든 일을 자동화하고 다른 실무자들에 대한 공감대를 형성하는 데 경력의 전반부를 보냈습니다. Google 클라우드 애플리케이션 현대화 프로그램(CAMP)의 공동 작성자이며 DORA 핵심팀의 멤버입니다. 또한 DevOps Enterprise Guidebook의 저자입니다. Google에서 근무하기 전에는 Chef Software에서 즐거운 동료들과 멋진 소프트웨어를 개발했습니다.



존 스피드 마이어

존 스피드 마이어는 소프트웨어 공급망 보안 스타트업인 Chainguard의 보안 데이터 과학자입니다. 존이 진행한 연구 프로젝트의 주제로는 소프트웨어 공급망 보안, 오픈소스 소프트웨어 보안, 중국 군사력 성장에 대한 세계의 대응 등이 있습니다. 이전에는 In-Q-Tel, RAND Corporation, Center for Strategic and Budgetary Assessments 에서 근무한 바 있습니다. 존은 파디 랜드 대학원에서 정책 분석 박사 학위를, 프린스턴 공공국제정책 대학원에서 MPA를, 터프츠 대학교에서 국제 관계 학사 학위를 받았습니다.



카이위안 '프랭크' 수

카이위안 '프랭크' 수는 Google의 정량적 사용자 경험 연구원입니다. 카이위안은 로그 및 설문조사 데이터를 분석하여 Google Cloud 제품의 사용 패턴과 사용자 피드백을 파악하고 개발자를 위한 제품 우수성을 개선합니다. Google에서 근무하기 전에는 Microsoft에서 Azure 및 Power Platform 제품에 대해 수년간 정성적 및 정량적 사용자 연구를 수행했습니다. 워싱턴 대학교에서 인간 중심 설계 및 엔지니어링 석사 학위를 받았습니다.



네이슨 하비

네이슨 하비는 Google의 개발자 관계 엔지니어로, 팀이 비즈니스 성과에 맞게 기술을 조정하면서 잠재력을 실현하도록 지원하는 업무를 수행했습니다. 네이슨은 최고의 팀 및 오픈소스 커뮤니티와 협력하여 DevOps 및 SRE의 원칙과 관행을 적용하는 데 도움을 주었습니다. 네이슨은 2020년 O'Reilly의 *97 Things Every Cloud Engineer Should Know*를 공동 편집하고 기고했습니다.



토드 콜레자

토드 콜레자는 Google의 사용자 경험 연구원으로서, 오늘날 소프트웨어 엔지니어가 작업하는 방식과 미래에는 이러한 작업 방식을 어떻게 개선할 수 있는지에 대해 연구합니다. 오리건 주립대학교에서 컴퓨터 공학 박사 학위를 받았습니다.

10

방법론

연구 설계

이 연구는 **추론적 예측**이라는 단면적 이론 기반 설계를 사용하며, 이는 오늘날 비즈니스 및 기술 연구에서 수행되는 가장 일반적인 설계 유형 중 하나입니다. 추론적 예측 설계는 순수한 실험용 설계가 실용적이지 않거나 불가능한 경우에 사용됩니다.

대상 모집단 및 샘플링

이 설문조사는 기술 및 혁신, 특히 DevOps에 익숙한 사람들과 관련되거나 긴밀하게 협력하는 실무자와 리더를 대상으로 했습니다. 이메일 목록, 온라인 프로모션, 온라인 패널, 소셜 미디어를 통해 설문조사를 홍보하고 사람들에게 설문조사를 네트워크와 공유하도록 요청했습니다(즉, 눈덩이 표집).

잠재 구성 생성

가능한 한 이전에 검증된 구성을 사용하여 가설과 구성을 공식화했습니다. 이론, 정의, 전문가 의견을 기반으로 새로운 구성을 개발했습니다. 그런 다음 설문조사에서 수집된 데이터가 신뢰할 수 있고 유효할 가능성이 높은지 확인하기 위해 의도를 명확히 하기 위한 추가 조치를 취했습니다.¹

저조한 성과자와 우수한 성과자의 차이 계산

'비교 방법' 섹션에서는 4가지 배포 성과 측정항목에 따라 저조한 성과자와 우수한 성과자를 비교합니다. 이 방법은 간단합니다. 배포 빈도를 예로 들어보겠습니다. 성과가 우수한 클러스터는 주문 시 배포합니다(예: 하루에 여러 번). 하루에 평균 4회의 배포를 수행한다고 하면 1년에 1,460회(4 * 365)의 배포가 됩니다. 반대로 저조한 성과자는 월 1회에서 6개월에 1회 사이의 빈도로 배포하며 평균 배포 빈도는 3.5개월에 1회로, 1년에 평균 약 3.4회(12/3.5) 배포하는 꼴입니다. 비율(우수한 성과자 1,460회 배포/저조한 성과자 3.4회 배포)을 계산하면 우수한 성과자가 저조한 성과자보다 417배 더 많다는 결과를 얻게 됩니다. 이 접근 방식은 다른 개발 성과 측정항목에도 일반적으로 적용됩니다.

¹ Churchill Jr, G. A. "A paradigm for developing better measures of marketing constructs," Journal of Marketing Research 16:1, (1979), 64-73.

통계 분석 방법

클러스터 분석

두 클러스터링 솔루션이 '비교 방법' 섹션에 잘 나와 있습니다. 서로 다른 클러스터 솔루션이 데이터에 얼마나 잘 맞는지 평가하기 위해 Ward의 병합적 방법² 버전과 함께 계층적 군집화를 사용했습니다.

제시된 첫 번째 클러스터링 결과의 경우 배포 빈도, 리드 타임, 서비스 복원 시간, 변경 실패율에 대한 응답 클러스터를 찾았습니다. 올해는 클러스터 수를 결정하기 위해 30개의 서로 다른 지표를 사용하여 14개의 서로 다른 계층적 군집화 솔루션을 평가한 결과 3개의 클러스터를 찾았습니다³.

제시된 두 번째 클러스터 분석은 방법론적으로는 첫 번째와 동일하지만 데이터에서 여러 측정기준을 통해 배포되었습니다. 우리는 처리량(배포 빈도와 리드 타임의 조합), 운영 성과(

신뢰성), 안정성(서비스 복원 시간과 변경 실패율의 조합)에서 공통된 응답 패턴(즉, 클러스터)을 찾고 싶었습니다. 또한 다양한 클러스터링 알고리즘을 탐색하여 결과가 접근 방식에 얼마나 민감한지 확인했습니다. 우리가 알고 있는 한 이러한 민감도를 정량화하는 확립된 방법은 없지만, 나타나는 클러스터는 유사한 특성을 갖는 경향이 있었습니다.

평가 모델

분석을 수행하기 전에 varimax 회전을 사용하는 주요 구성 요소 분석과 탐색적 요인 분석을 사용하여 구성을 식별했습니다.⁴ 평균 분산 추출(AVE), 상관관계, 크론바흐 알파 계수⁵, rhoA_A, heterotrait-monotrait ratio^{5,7}, 합성 신뢰도를 사용하여 수렴, 확산적 타당도 및 신뢰도에 대한 통계적 검정을 확인했습니다.

² Murtagh, Fionn, and Pierre Legendre. "Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion?." *Journal of classification* 31.3 (2014): 274-295.

³ Charrad M., Ghazzali N., Boiteau V., Niknafs A. (2014). "NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set.", *Journal of Statistical Software*, 61(6), 1-36., "URL <http://www.jstatsoft.org/v61/i06/>"

⁴ Straub, D., Boudreau, M. C., & Gefen, D. (2004). Validation guidelines for IS positivist research. *Communications of the Association for Information systems*, 13(1), 24.

⁵ Nunnally, J.C. *Psychometric Theory*. New York: McGraw-Hill, 1978

⁶ Hair Jr, Joseph F., et al. "Partial least squares structural equation modeling (PLS-SEM) using R: A workbook." (2021): 197.

⁷ Brown, Timothy A., and Michael T. Moore. "Confirmatory factor analysis." *Handbook of structural equation modeling* 361 (2012): 379.

구조방정식 모델링

상관관계 기반 구조방정식 모델링인 부분 최소 제곱(PLS) 분석⁸을 사용하여 구조방정식 모델(SEM)을 테스트했습니다.

두 번째 클러스터 모델 분석

클러스터 구성의 예측 요소를 파악하기 위해 다항 로지스틱 회귀를 채택했습니다.⁹ 이 경우에는 3개 이상의 수준으로 된 정렬되지 않은 범주형 데이터로 된 클러스터 구성을 예측하고자 했기 때문에 이 접근 방식을 사용했습니다. 클러스터 구성이 예측한 결과를 이해하기 위해서는 각 결과(번아웃, 무계획적 업무, 조직 성과)에 대해 선형 회귀를 사용했습니다.

⁸ Hair Jr, J. F., Hult, G. T. M., Ringle, C. M., & Sarstedt, M. (2021). "A primer on partial least squares structural equation modeling (PLS-SEM)." Sage publications.

⁹ Ripley, Brian, William Venables, and Maintainer Brian Ripley. "Package 'nnet'." R package version 7.3-12 (2016): 700.

11

추가 자료

DORA Community에 참여하여 소프트웨어 배포 및 운영 성과 개선을 주제로 논의하고 배우며 공동작업해 보세요.

<http://dora.community>

4가지 핵심 측정항목 자세히 알아보기

<https://goo.gle/four-keys>

DevOps 역량에 대한 자세한 정보 찾기

<https://goo.gle/devops-capabilities>

엔터프라이즈 가이드북을 통해 조직에서 DORA 관행을 구현하는 방법 자세히 알아보기

<https://goo.gle/enterprise-guidebook>

사이트 안정성 엔지니어링(SRE)에 대한 리소스 찾기

<https://sre.google>

<https://goo.gle/enterprise-roadmap-sre>

DevOps 빠른 점검 받기

<https://goo.gle/devops-quickcheck>

DevOps 연구 프로그램 살펴보기

<https://goo.gle/devops-research>

Google Cloud DevOps 어워즈 수상자 eBook을 읽고 DORA 관행을 구현한 다른 회사로부터 배우기

<https://goo.gle/devops-awards>

Google Cloud 애플리케이션 현대화 프로그램 (CAMP) 알아보기

<https://goo.gle/3daLa9s>

데이터 및 DORA 측정항목을 활용하여 기술 프로세스 혁신

<https://goo.gle/3Doh8Km>

백서 읽기: “The ROI of DevOps Transformation: How to quantify the impact of your modernization initiatives” by Forsgren, N., Humble, J., & Kim, G. (2018).

<https://goo.gle/3qECllh>

도서 읽어보기: *Accelerate: The science behind devops: Building and scaling high performing technology organizations*. IT Revolution.

<https://itrevolution.com/book/accelerate>

SLSA(Supply Chain Levels for Software Artifacts) 프레임워크 자세히 알아보기

<https://slsa.dev>

NIST SSDF(Secure Software Development Framework) 자세히 알아보기

<https://goo.gle/3qBXLWk>

DevOps 문화: 웨스트럼 조직 문화 자세히 알아보기

<https://goo.gle/3xq7KBV>

Open Source Security Foundation 자세히 알아보기

<https://openssf.org/>

in-toto 자세히 알아보기

<https://in-toto.io/>

NTIA.gov의 Software Bill of Materials 자세히 알아보기 <https://www.ntia.gov/SBOM>

Cybersecurity: Federal Response to SolarWinds and Microsoft Exchange Incidents

<https://www.gao.gov/products/gao-22-104746>

CI/CD에 포함된 애플리케이션 수준 보안 스캔 자세히 알아보기

<https://go.dev/blog/survey2022-q2-results#security>

마지막으로 이전 State of DevOps Report를 확인하세요. 모든 보고서가 다음 페이지에 등재되어 있습니다. <https://goo.gle/dora-sodrs>:

<https://goo.gle/dora-sodrs>:

[2014 Accelerate State of DevOps Report](#)

[2015 Accelerate State of DevOps Report](#)

[2016 Accelerate State of DevOps Report](#)

[2017 Accelerate State of DevOps Report](#)

[2018 Accelerate State of DevOps Report](#)

[2019 Accelerate State of DevOps Report](#)

[2021 Accelerate State of DevOps Report](#)

