

A-SPICE 준수: 시스템 개발 시 직면할 수 있는 위험과 대응방안

Automotive SPICE(A-SPICE)을 준수하려면 경험이 필요하고 다양한 도전과제가 제기된다. 그 중 시스템 개발의 가장 일반적인 문제점 두 가지를 설명하고, 프로세스 개선 프로젝트에서 발생하는 문제의 해결 방안을 공유한다. 또한 프로세스 개선뿐만 아니라 시스템 엔지니어링의 함정을 피하는 방안을 공유한다.

글 | 볼커 레만(Volker Lehmann) 이사
메소드파크(Method Park)

한컴MDS는 지난 5월 29일 서울 삼성동 코엑스 그랜드볼룸에서 '자동차 소프트웨어(SW) 개발자 컨퍼런스 2018'을 개최했다. 올해는 독일 메소드파크(Method Park)의 볼커 레만(Volker Lehmann) 이사가 기조연설을 했다. 레만 이사는 자동차 SW 개발 프로세스 국제표준 A-SPICE의 인증기관 인탁스(iNTACS) 자문위원이기도 하다. 그는 A-SPICE 준수의 중요성을 주제로 자동차 시스템 개발 시 직면할 수 있는 위험과 대응방안에 대해 소개했다. 이 글은 레만 이사의 기조연설을 발췌, 정리한 것이다.

[요약]

- Automotive SPICE®를 도입하면 체계적이고 전문적인 시스템 및 소프트웨어 엔지니어링을 구축할 수 있다.
- 자동차 도메인의 시스템 오버뷰가 없으면 불일치와 결함이 발생할 위험이 높다.
- 성공적인 프로세스 개선을 위해서는 기술 전문가의 개입과 경영진의 높은 관심이 중요하다.
- 프로세스 관리 도구는 일관되고 적용 가능한 프로세스 개발을 강력하게 지원한다.
- Automotive SPICE® 전문가 커뮤니티가 성장하고 있으며, 여러분은 혼자 아니다.

나는 인탁스(iNTACS) 자문위원회 위원으로서 A-SPICE 개발, 프로세스, 트레이닝, 실러버스(Syllabus) 작성 등에 관여하고 있다. 주로 교육 및 평가(Assessment) 분야에서 코칭하는 업무를 담당하고 있다. 또한 오랜 경험을 공유하고 A-SPICE를 이해할 수 있도록 독일에서 커뮤니티 지원 일도 하고 있다.

내가 소속된 메소드파크(Method Park)는 자동차 엔지니어링 프로세스 관리 분야 세계 1위 기업이다. 메소드파크는 프로세스 관리 툴을 고객사에 제공해 프로세스를 개발하고, 도입하고, 유지 관리할 수 있도록 지원하고 있다.

복잡성 증가

왜 우리는 표준과 규격에 대해서 이야기해야 할까? 우리 앞에는 많은 도전과제가 산적해 있고 시장은 복잡해지고 있다. 자율주행 차량이 대두되면서 보안과 안전은 더욱 중요해졌다. 때문에 더 많은 기능이 자동차에 통합될 수밖에 없다. 반면 제어장치는 줄어들고 개발은 점점 어려워지고 있다. 또한 예산은 줄고 개발시간은 짧아지고 있다. 개발의 세계화라는 과제도 대두되고 있다. 완성차 업체뿐만 아니라 그 협력업체들도 여러 지역, 여러 국가

에서 활동하고 있다. 때문에 일하는 방식, 경험, 정보를 서로 공유할 수 있어야 한다. 또한 잦은 인력 변동과 전문가 부족도 프로젝트의 복잡성을 증가시키는 원인 중 하나이다.

직원이 이직을 하면 지식도 함께 사라진다. 그 지식을 보존하기 위해서는 개발 주기, 문서와 관련된 프로세스를 통해서 대처해야 한다. 그래야 개발 과정에서 조정(Alignment)이 가능하고, 여러 가지 산출물이나 개발 그 자체를 공유할 수 있다. 이 경우 표준 프로세스 모델이 상당히 도움이 된다.

그런 측면에서 A-SPICE가 어떻게 도움을 줄 수 있을까? A-SPICE는 아주 중요한 목표를 가지고 있다. 우선, 체계적이고 전문적인 시스템 및 소프트웨어 엔지니어링을 확립할 수 있도록 돕는다. 또한 평가(Rating)가 가능하며, 시스템/소프트웨어 복잡성을 관리할 수 있도록 한다. 즉 어떤 제품을 개발하는데 있어서 제대로 테스트되고, 안정적이며, 결함을 최소화할 수 있도록 한다. 또 다른 부분은 A-SPICE를 통해 실제로 필요하고 적합한 프로세스를 정의할 수 있다는 것이다.

세계화와 관련해서는, 전체 조직 내의 모범 사례(Best practices)를 기반으로 프로세스를 공유함으로써 모든 개발자가 똑같은 규칙 하에 개발하고, 똑같은 내용에 대해서 대화할 수 있다는 장점이 있다.

다음은 공급망(Supply chain)에 대해 생각해봐야 한다. 결국 우리는 공급망의 일부가 된다. 우리가 고객사일 수도 있고, 어떤 경우에는 1차, 2차, 3차 협력업체가 될 수도 있다. 어떤 경우는 상관없다. 공급망 전반에 걸쳐 공통 프로세스나 정의된 인터페이스를 갖춘다면 제품 개발과정이 좀 더 원활해진다. 또한 공급망 전반의 개발상황을 모니터링할 수 있다. 이것을 강력하게 지원하는 것이 바로 A-SPICE 프로세스이다.

A-SPICE의 역사

A-SPICE는 사실 새로운 것이 아니다. 1990년대 SPICE의 기초가 마련되었다. SPICE는 Software Process Improvement

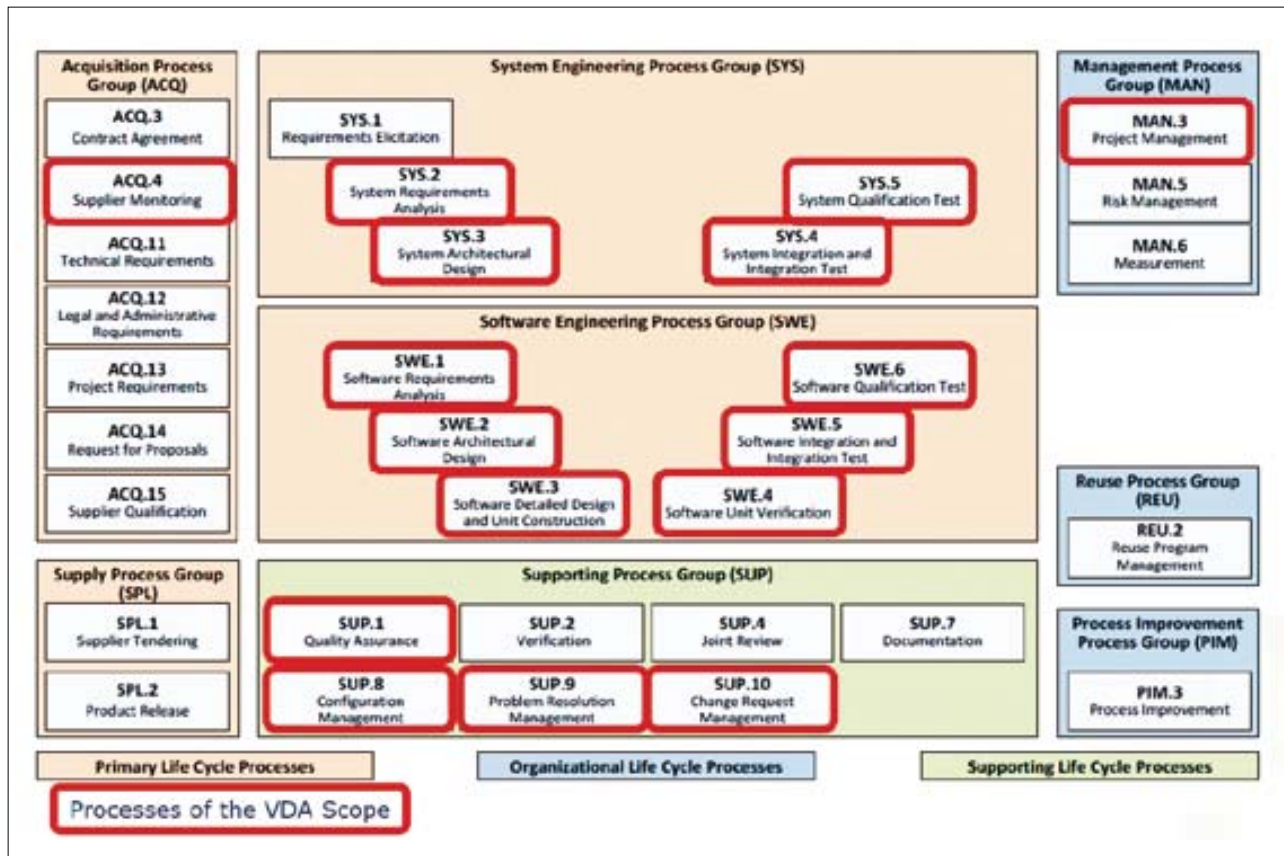
and Capability Extermination의 약자이다. 2003년부터 2005년 사이에 SPICE에서 파생된 A-SPICE가 제정되었다.

A-SPICE는 독일 자동차협회(VDA)가 관장하며 현재 3.0 버전이 릴리즈 되었다. 새로운 문서 Automotive SPICE Guideline이 2017년 공개됐는데, 표준을 어떻게 이해하고 규격을 어떻게 이행해야 하는지 가이드가 담겨 있다.

A-SPICE에는 크게 두 가지 내용이 담겨 있다. 우선, 프로세스들이 정의되어 있다. 이 프로세스들은 개발에 있어서 여러 부분을 커버한다.



메소드파크(Method Park)는 자동차 엔지니어링 프로세스 관리 분야 세계 1위 기업이다.



A-SPICE 프로세스가 커버하는 부분

역량 레벨

A-SPICE는 레벨 0에서 레벨 5까지 역량 레벨이 있다. 각각의 프로세스가 각 레벨의 요구사항을 충족하는지 여부에 따라 평가된다.

가장 중요한 것은 A-SPICE를 어떻게 도입하느냐이다. 고객사와 대화를 나눌 때나 프로세스 작업을 할 때 일반적으로 직면하게 되는 두 가지 도전과제를 발견했다. 첫 번째는 제품을 개발하면서 A-SPICE 요구사항을 충족시키는 것이다. 두 번째는 A-SPICE에 맞춰 프로세스를 개선하는 것이다. 이는 프로젝트 관리가 아니라 프로세스 관리와 관련된 것이다. 이 과정에서 여러 가지 어려움이 있을 수 있기 때문에, 도전과제로 제시된 것이다. 이 두 가지 도전과제에 대해서 상세히 소개하고자 한다.

도전과제

첫 번째 도전과제는 제품 개발에 관한 것이다. 즉 시스템 엔지니어링 부분이다. 개발자라면 시스템 엔지니어링을 잘 알고 있을 것이다. 우선, 소프트웨어 개발의 경우 분석,

검증, 배포 등 반복해야 하는 과정이 복잡하다. 또한 동시에 모든 항목이 하드웨어와 정확하게 피팅(Fitting)되어야 한다. 이와 마찬가지로 소프트웨어와 피팅된 하드웨어는 시스템의 일부가 된다. 하드웨어는 사이클이 많지는 않지만 하드웨어 쪽에도 변경은 있다. 여기에 메커니컬(Mechanical) 부분까지 합쳐지면 우리가 개발하고자 하는 시스템이 되는 것이다. 중요한 것은 인터페이스와 동기화이다. 프로젝트 전체 라이프사이에 걸쳐 인터페이스와 동기화에 대해서 관심을 가져야 한다.

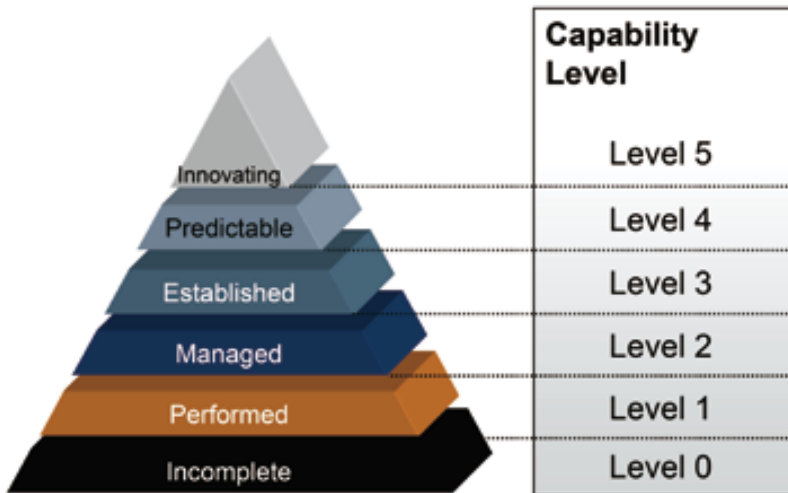
메커니컬, 하드웨어, 소프트웨어 각 영역만을 보게 되면, 많은 업체들이 특별한 프로세스가 없어도 개발이 잘 진행되고 있는 듯 보인다. 사실이다. 그렇다면 무엇이 문제일까? 문제는 세 가지가 제대로 결합이 안 된다는 데 있다. 각각은 다 잘 진행된다. 그런데 메커니컬, 하드웨어, 소프트웨어가 결합된 시스템 레벨에서 프로젝트를 평가할 때도 잘 되고 있을까? 이것이 우리가 직면한 도전과제라 할 수 있다.

각각의 영역이나 분야에 대해서는 어느 정도 이해하고 있지만, 모든 내용이 취합될 경우 시스템 혹은 그 상위에서 현황을 파악할 수 있는 오버뷰가 없으니 시스템이 요구사항대로 개발되고 있는지 알 수가 없다.

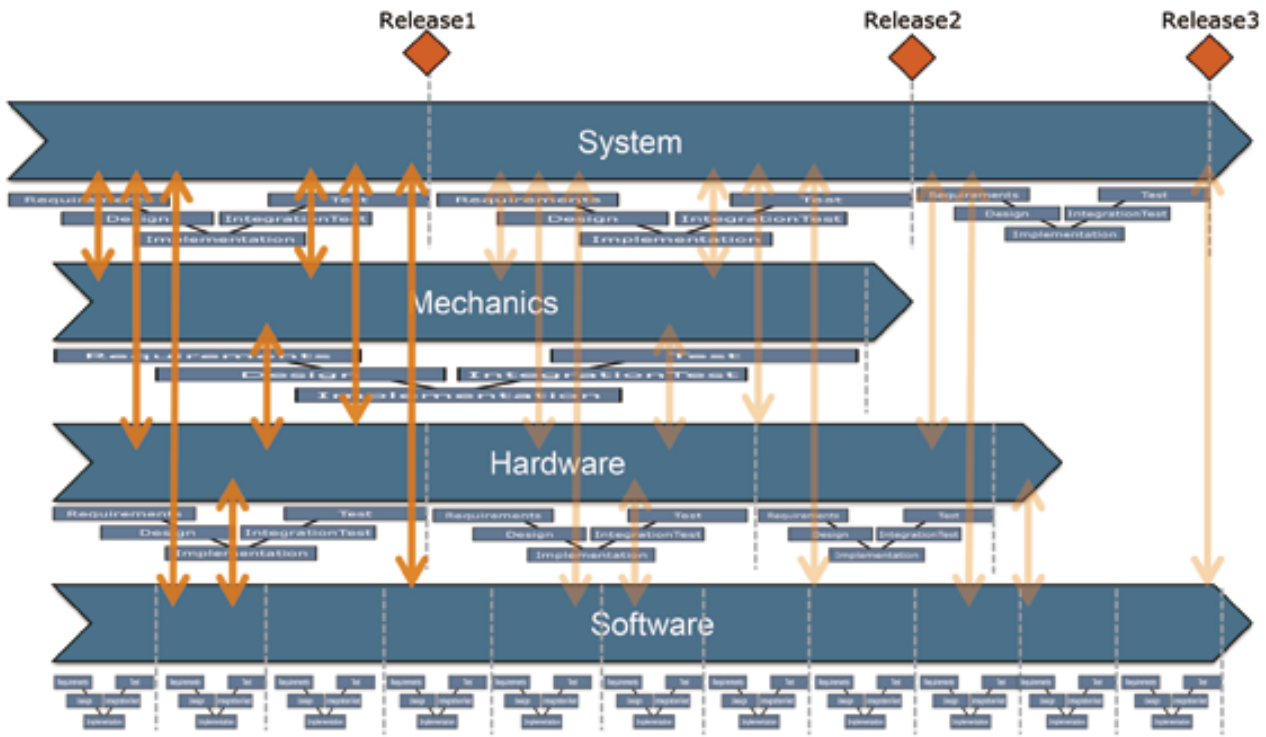
즉 각 영역별 인터페이스가 제대로 관리되지 않는다는 데 문제가 있다. 심지어 어떤 인터페이스가 정의되어 있는지 모르는 경우가 대부분이다. 메커니컬, 하드웨어, 소프트웨어가 제대로 작동하도록 요구사항들이 정렬되어 있는지, 그리고 하드웨어 요구사항과 소프트웨어 요구사항들이 제대로 연결되어 있는지 모르기 때문에 결국 알 수 없는 문제가 발생한다.

형상 관리(Configuration Management)는 어떤가? 각 도메인별로는 잘 이루어지고 있다. 나름의 워크플로로 자신들의 산출물을 잘 만들어내고 있다. 하드웨어 엔지니어들은 자신이 하는 일을 잘 알고 있고 하드웨어를 잘 만들어낸다. 메커니컬 쪽이나 소프트웨어 쪽도 마찬가지다. 소프트웨어를 위한 빌드 프로세스가 잘 되어 있고 규칙들도 잘 마련돼 있다. 그래서 적합한 품질의 소프트웨어가 공급되기도 한다. 하지만 이 세 가지가 제대로 조화롭게 잘 진행되는지는 여전히 의문이다. 하드웨어에 맞는 소프트웨어를 어떻게 찾고, 소프트웨어에 맞는 하드웨어는 어떻게 찾느냐의 문제가 있다.

시스템 레벨에서는 테스트와 관련된 문제가 발생할 수 있다. 프로젝트의 규모나 복잡도에 따라서 수많은 테스트 케이스가 만들어지는데, 이 경우 프로젝트에 참여하는 사람들이 어떤 부분을 테스트하는지조차 모르는 경우가 있다. 자신들이 시스템을 테스트하는지, 소프트웨어를 테스트하는지, 소프트웨어와 하드웨어를 조합해서 테스트하는지 모르는 것이다. 그러다 보니 테스트 커버리지가 제대로 지켜지지 않는 문제가 발생한다. 이는 결국 추가적인 문제로 이어진다. 시스템과 소프트웨어 등 요구사항에 대응하는 테스트를 하고 결과를 확인한다 해도 모든 사항을 고려해서 테스트 했는지는 알 수 없다. 다시 말해, 충분한 테스트 결과에도 불구하고 전체 시스템이 제대로 작동할지 장담할 수 없다.



ASPICE의 프로세스 역량 레벨



제대로 된 시스템을 만들기 위해서는, 소프트웨어/하드웨어/메커닉 간의 정확한 인터페이스 정의와 동기화 관리를 할 수 있는 프로세스 관리 시스템이 반드시 필요하다.

아키텍처 레벨에서도 마찬가지다. 통합 시험을 통해 일부 인터페이스는 확인할 수 있다. 그러나 모든 인터페이스가 커버되는지는 알 수 없다. 시스템에 대한 오버뷰가 없기 때문이다. 소프트웨어에서 예상할 수 없었던 결함이 시스템에 잠재적으로 남아서 중대한 고장으로 발현될 수 있다는 의미이다.

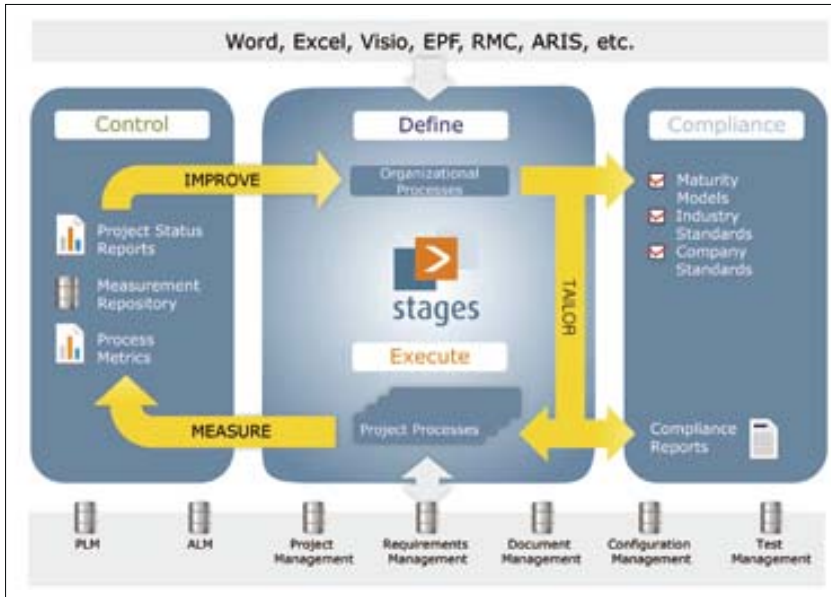
앞서 언급했지만, 시스템이나 소프트웨어의 복잡성이 증가하고 있는 데 반해 개발 시간과 예산은 항상 부족하다. 결국 처음부터 제대로 해야 한다. 그럼, 어떻게 제대로 할 수 있을까?

오랜 경험을 토대로 말할 수 있는 것은, A-SPICE 기반의 프로세스 개선이 하나의 답이 될 수 있다는 것이다. A-SPICE의 경우, 이론적으로 제대로 돌아가는지 뿐만 아니라, 실제로 제품이 요구 기준에 따라 개발되어 동작하는지 확인하는 데 중점을 두고 있다.

A-SPICE 기반의 프로세스 엔지니어링과 프로세스 개선은 더욱 중요하다. 프로세스 개선(Process Improvement, PI)과 관련해서는 당연히 그 나름의 도전과제가 있다. 그 중에 하나가 바로 프로세스 개선과 관련된 상아탑 문제라고 할 수 있다. 대부분의 프로세스 팀은 경험이 많다. 어떤 표준을 준수해야 하는지를 이해하고 프로세스 정의를 위한 방법론도 잘 알고 있다. 경험이 많기 때문에 표면적으로는 문제가 없는 듯한 프로세스 환경을 구축해 놓았을 수 있다. 하지만 가장 큰 문제는, 정의된 프로세스가 실질적으로 프로젝트와 일치하지 않는 경우가 많다는 데 있다. 개념적(이론적)으로는 잘 돼 있지만 회사나 실제 현장에서 필요한 요구를 간과하는 경우가 있기 때문에 실제 프로젝트가 사용되지 못하는 것이다. 즉 현실을 고려하지 않았기에 프로세스는 있으나 프로젝트의 수행과는 일

치가 안 되고, 따라서 프로젝트 팀원들에게 어필이 안 되는 것이다. 선반 위에 예쁘게 진열해 놓은 장식일 뿐 실제로 사용되지 않는 프로세스가 된다는 것이다. 수많은 사람들의 시간과 돈을 투입해서 만든 프로세스가 무용지물이 된다는 것, 이것이 우리가 가장 조심해야 할 문제이다.

또 다른 도전과제는 프로세스 개선(PI)을 위한 프로젝트를 야심차게 시작했어도, 곧 점진적으로 사라진다는 것이다. 프로세스 개선(PI)을 하려면 경영진의 적극적인 참여가 있어야 한다. 그런데 PI에 대해서는 회사의 지원이 점차 축소되는 경우가 많다. 성과를 바로 확인하기 어려운 PI 프로젝트는, 경영진 입장에서 관심이 별로 없기 때문에 결과적으로 예산이 삭감되고 인력이 감축된다. 그래서 프로세스 팀은 기본적으로 최소 단위가거나 아니면 아예 사람이 없는 경우도 있다. PI 프로젝트는 특히 경영진이 직



프로세스 개선은 단기간에 달성할 수 있는 단순한 것이 아니며, 성공적인 프로세스 개선 활동을 위해서는 적합한 프로세스 관리 시스템의 구축이 필요하다.

접 모니터링하고 리포트를 요구하면서 지속적인 투자가 이루어져야 한다. 하지만 쉽지 않은 게 현실이다. 결국 초창기의 대규모 투자에도 불구하고 결과물이 존재하지 않거나 수준이 낮은 경우가 많다. 따라서 경영진의 직접적인 관여가 반드시 필요하고 프로젝트 체계가 제대로 잡히는 것이 중요하다.

또 다른 측면으로 언급하고 싶은 것은, 이런 문제들은 분명히 해결 가능하다는 것이다. 전문가들을 프로세스 팀에 영입해서, 프로세스 팀원들과 코치들을 프로세스 개선 활동에 좀 더 적극적으로 개입시켜야 한다. 전문가들이 보유하고 있는 충분한 경험과 노하우, 기술 지식을 활용하면 회사에 최적화된 프로세스가 적용될 것이다.

프로세스 개선은 단순한 예외 활동이 아니다. 프로세스를 특정하고 개발, 구현해야 하며 검증이 되어야 한다. 또한 프로세스 관리 체계를 구축해서 프로세스가 제대로 돌아가는지도 확인해야 한다. 이를 위해서는 프로세스 관리를 위한 전문 시스템이 구축되어 지속적인 프로세스 자체의 유지관리가 필요하다. 이를 고려하면, 어쩌면 일반적인

개발 프로젝트보다 많은 공수가 들어간다고 할 수 있다. 개발 프로젝트에서 고려되는 사항들이 PI 프로젝트에서도 적용되는 것이다. 즉 프로세스를 개선하기 위해서는 프로젝트 구조(Project structure)로 접근이 필요하다.

프로젝트 구조로 접근을 하려면 리포팅(reporting)이 있어야 한다. 누가 어떤 사람에게 리포팅을 해야 하는지 정의해야 한다. 즉 누가 A-SPICE를 구현할지 확인하고, 이러한 구현이 어떻게 진행되었는지 리포팅되어야 한다. 이를 통해 경영진은 어떤 이슈가 어디에 발생했는지 확인하고, 제대로 구현이 진행되고 있는지 점검해야 한다.

프로세스 관리 도구

성공적인 프로세스 개선 활동을 위해서 또 무엇이 필요할까?

반드시 필요한 것 중에 하나는 프로세스 개선 활동을 충분히 지원할 수 있는 적합한 프로세스 관리 도구를 사용하는 것이다. 어떤 도구를 사용하든 간에 프로세스 개

선 활동을 위한 도구는 워드나 엑셀, RMC 등 다양한 방법으로 정의된 프로세스 정보들을 수집하여 표현할 수 있어야 한다. 도구 내에 프로세스 정의, 조직에 대한 표준 등을 정의해서 관리할 수 있어야 한다. 이 경우 자신이 준수한 표준을 제대로 준수했는지 도구를 통해 확인할 수 있다. 프로세스 관리 도구에서 다루는 표준은 사내 표준을 포함하여 ISO 26262, 보안과 같은 ISO 표준, A-SPICE 등이 있을 수 있다. 즉 프로세스 관리 도구를 통해 표준의 준수 여부를 확인하는 동시에 프로젝트 내에서 활용할 수 있도록 관련된 프로세스를 구현해야 한다.

프로세스는 프로젝트를 통해서 활용된다. 따라서 프로세스 관리 도구는 프로젝트 관리를 위해 구축된 시스템(제품 라이프 사이클 매니지먼트, 애플리케이션 라이프 사이클 매니지먼트, 컨피규레이션 매니지먼트 등)과의 연계가 필수적이다. 프로덕트/프로젝트 관리 도구와의 연계를 통해, 프로젝트가 정의된 프로세스대로 수행되고 있는지 모니터링하면서 프로세스와 조직의 개선 방향을 도출할 수 있기 때문이다. 뿐만 아니라, 프로세스 관리 도구를 활용하면 프로세스 내 일관성 확보에 큰 도움이 된다. 다양한 프로세스 버전 관리도 가능하다.

이처럼 프로세스 관리를 위한 적합한 도구 활용을 통해 해당 프로젝트에서 제품 품질과 업무 효율성을 향상시킬 수 있는 실제 프로세스를 가능케 한다.

마지막으로 강조하고 싶은 것은, A-SPICE는 하나의 도구라는 점이다. 체계적인 시스템 엔지니어링을 위한 실용적인 도구로서 도움을 줄 수 있다. 또 시스템 오버뷰가 없으면, 프로젝트에서 리스크가 많고 일관성이 없어질 수밖에 없다. 또한 성공적인 프로세스 개선 프로젝트를 위해서는 경영진의 관심과 전문가들이 필요하다는 점을 강조하고 싶다. 적합한 프로세스 관리 전문 도구의 활용도 필수라고 할 수 있다.