

소규모의 소프트웨어개발단위들에서 공정개선을 위한 한가지 방법

문일남, 량기선, 김철성

경애하는 최고령도자 김정은동지께서는 다음과 같이 말씀하시였다.

《모든 부문, 모든 단위에서 품질공학수법이 가지는 중요성을 깊이 인식하고 자체의 실정에 맞게 품질공학연구사업을 힘있게 벌리며 그 성과들을 생산에 제때에 받아들이기 위한 사업을 계획적으로 밀고나가야 합니다.》

오늘날 소프트웨어산업은 가장 급속히 장성하는 분야들중의 하나이며 소규모의 소프트웨어개발단위들이 정보산업분야에서 중요한 역할을 하고있지만 이런 개발단위들에서의 소프트웨어공정개선(SPI: Software Process Improvement)을 위한 연구자료는 많지 못하다.

소프트웨어개발에서 공정이란 입력을 출력으로 변환할 때 호상작용하는 활동들의 모임이며 소프트웨어개발은 일반적으로 예정된 일정을 지키면서 최소한 적은 비용으로 최대한 질높은 소프트웨어를 생산하는것 즉 일정, 비용, 품질의 최적화를 실현할수 있는 공정을 가질것을 요구하고있다.[2, 4]

대부분의 소프트웨어개발단위들은 소규모(50명미만)이고 대부분의 프로젝트들도 소규모(20명미만)이며 때로는 극소규모(5명미만)인 경우도 많다. 잘 정의된 공정들은 위에서 언급한 최적화문제들을 해결하는데 도움이 될수 있으며 이를 위한 SPI는 대규모의 단위들만이 아니라 소규모의 단위들에서도 경쟁상우위를 차지하기 위한 전략으로 리용될수 있다.

한편 최근 몇년사이에 많은 소프트웨어개발단위들이 소프트웨어공정개선에 관심을 두고있다. [1]

소프트웨어를 개발하는 공정의 품질을 개선하면 그 소프트웨어의 품질을 개선할수 있다는것이 증명된 1980년 후반기부터 SPI에 대한 연구가 많이 진행되였지만 그것은 주로 대규모 개발단위들에 관련된것이고 소규모의 개발단위들과 관련한 정보는 그리 많지 못하다.[2]

소프트웨어공정을 개선하는데서 첫 걸음은 현재의 공정을 모형화하는것이며 이 첫 걸음을 떼는 좋은 방도는 CMMI와 SPICE, ISO의 품질표준 ISO 9001과 같은 잘 알려진 SPI 모형에 기초하여 공정개선노력을 뒤받침하는것인데[3] 이런 SPI모형의 수법들은 대체로 수백명의 종업원들을 가진 대규모의 단위들에 맞게 설계되어있으며 따라서 소규모의 단위들에 적용하기도 어렵고 이런 소규모의 개발단위들은 흔히 잘 알려진 SPI를 수행할만 한 지식과 자원도 가지고있지 못하다.[2]

또한 소규모의 단위들에 맞게 개발된 방법과 모형들이 몇가지 있지만 그것들은 보통 SPI 모형에 기초하여 개발된것으로 하여 그것들을 적용하자면 기초로 되고있는 SPI모형에 대한 지식이 필요하게 된다.

그러므로 소규모의 소프트웨어개발단위들의 특성에 맞게 SPI모형에 대한 지식을 요구

하지 않으면서도 해당 단위의 업무목적에 부합되고 중요한 소프트웨어공정들에 초점을 둘 뿐아니라 최적이면서 유연하고 리용하기 쉬운 SPI모형을 찾아내는것이 필요하다.

이런 문제점들을 고려하여 논문에서는 소규모의 소프트웨어개발단위들에서 공정을 개선하는데 리용할수 있는 한가지 간단하면서도 실용적인 경량SPI방법을 제안하였다.

1. PISKO기법에 기초한 경량SPI방법

여기서는 소프트웨어공정을 모형화하려고 할 때 해당 단위의 종업원들의 의견과 경험에 중점을 두는 PISKO기법[2]에 기초한 한가지 SPI방법을 제안한다.

종전의 방법은 6개의 단계를 거치게 되어있지만 여기서는 4개의 단계 즉 벽도해기법에 의한 공정모형화, 공정서술과 분석, 문제식별, 공정의 개선방법을 제안한다.

① 벽도해기법에 의한 공정모형화

이 단계에서는 해당 단위의 일군들(5명정도)과 소프트웨어공학전문가들(2명정도)이 참가하며 일군들은 실제적인 공정을 모형화하고 전문가들은 이 활동들을 감시하고 기록한다. 우선 참가자들이 함께 토론하여 각이한 공정마디들에 대한 종이표식들을 벽도해에 붙이는 방법으로 진행된다. 공정마디들을 결정할 때 중요한것은 모든 종업원들이 자기 의견을 내놓도록 하는것과 마지막에 공정마디들과 그 이름에 관하여 모두가 합의를 이룩하는것이다. 공정마디들을 식별한 다음에는 공정의 진행상태를 보여줄수 있게 그것들을 련결한다. 이 첫 모형화단계의 결과가 바로 해당 단위의 현재의 상태를 보여주는 벽도해이다.

② 공정서술과 분석

모형화가 끝나면 벽도해에 토대하여 공정마디들을 본문으로 서술한 전자문서로 바꾼다. 이때 중점은 공정의 현재상태에 대하여 알기 쉽고 포괄적인 문서화를 하는데 둔다. 그리고 이 공정서술문서에 대해 해당 단위의 대표들이 검사하고 확인한 다음 검사된 문서에 대해 소프트웨어공학전문가들이 분석한다.

③ 문제식별

이 단계에서는 해당 단위의 일군들과 소프트웨어공학전문가들이 공동으로 공정서술문서와 분석자료를 놓고 개선해야 할 문제들을 식별한다. 여기서는 공정이 명확히 서술되고 정의되었는가, 너무 복잡하지 않는가, 중복되지 않는가, 쓸데없는 작업을 피할수 있지 않겠는가, 역할과 책임, 작업방식이 명확히 규정되어있는가 그리고 모임방식, 결심채택, 문서화가 적절한가 등에 주목을 돌린다. 중요한것은 문제점들에 대하여 해당 단위의 일군들과 소프트웨어공학전문가들사이에 합의를 이룩하는것이다.

④ 공정의 개선

이 단계에서는 공정을 명백히 하고 보다 효율적인것으로 되게 할뿐아니라 발견된 문제들을 제거하는데 중점을 둔다. 여기서는 해당 단위의 관리부서 일군들을 포함한 리해관계자들과 함께 공정서술문서를 갱신하면서 공정을 다듬어나간다. 이때 소프트웨어공학전문가들은 공정이 개선되는가를 세밀히 관찰하면서 조언을 준다. 또한 개선된 공정마디들에 작업방식의 실행을 책임진 사람뿐아니라 리해관계자, 개발자들을 모두 참가시킨다.

웹페이지를 통하여 혹은 벽에 걸어놓아 개선되어나가는 벽도해와 공정서술문서를 누구나 다 쉽게 알수 있도록 하는것이 중요하다.

2. 적용 사례

여기서는 위에서 설명한 경량SPI방법을 적용하여 컴퓨터망통신부분의 응용프로그램개발을 전문으로 하는 어느 한 개발단위(인원 약 20명)에서 자연스럽게 준수될수 있는 세련된 공정을 확립한 실례를 보여준다.

이 단위의 공정은 수요자들의 요구를 만족시키기 위한 새로운 기능들을 심의하는 개발모임으로부터 시작되어 수요자들에게 새로운 판본을 인도하는것으로 끝난다.

제안된 방법에 의하여 공정마디들이 19개로부터 10개로 다듬어져 공정이 단순해지고 준수하기 쉽고 효율적인것으로 되었으며 공정의 매 마디들에서 역할과 책임이 명백해지고 모임진행방식과 결심채택방식이 개선되었다. 그리고 단위의 종업원들이 문제식별단계에서 발견되었던 문제들이 더는 존재하지 않는다는 확신을 가지게 되었다.

또한 이 단위의 공정에 대한 모형화로부터 개선에 이르기까지 전체 개선작업에 투하된 자원은 인적자원뿐이었다. 구체적으로 본다면 공정을 모형화하고 분석한 다음 문제들을 식별하고 합의를 이룩하면서 단위의 일군들과 소프트웨어공학전문가들이 공동으로 투하한 시간이 연 86h, 소프트웨어공학전문가들이 공정을 서술한 전자문서를 작성하는데 투하한 시간을 합계하면 총 104h이 걸렸으며 모임에 리용된 시간은 총 24h이었다.

실제적인 개선작업은 녀달동안에 연 127h 걸렸고 실시 모임들에서 소비된 시간은 총 45h 이었다.(표 1, 2)

표 1. 공정모형화 및 분석, 문제식별에 리용된 자원

단 계	p-TC /명	p-R /명	h-TC /h	h-R /h	총계 /h
공정모형화	5	2	32	21	53
공정서술문서의 작성	0	2	0	6	6
공정서술문서의 검사 및 확인	2	0	12	0	12
문제식별	5	2	24	9	33
총 계			68	36	104

표 2. 개선에 리용된 자원

단 계	p-TC /명	p-R /명	h-TC /h	h-R /h	총계 /h
검사목록의 작성	2	2	7	2	9
공정의 다듬기	6	2	34	12	46
공정서술문서의 갱신	0	2	0	10	10
모임진행방식의 개선	2	1	9	6	15
결심채택방식의 개선	2	1	11	4	15
최종공정서술문서의 합의	6	2	17	15	32
총 계			78	49	127

표 1과 2에서 p-TC는 단위의 일군인원수, p-R는 소프트웨어공학전문가인원수, h-TC는 단위의 일군들이 리용한 연 시간수, h-R는 소프트웨어공학전문가들이 리용한 연 시간수, 총계는 단위의 일군들이 리용한 연 시간수와 소프트웨어공학전문가들이 리용한 연 시간의 합이다.

이와 같이 논문에서는 소규모의 소프트웨어개발단위들에서 공정의 개선에 리용할수 있는 한가지 간단하면서도 실용적인 경량SPI방법을 제안하고 적용실례를 통하여 제안된 방법의 효과성을 확증하였다.

참 고 문 헌

- [1] I. Richardson et al.; IEEE Software, **24**, 18, 2007.
- [2] Anu Valtanen et al.; Profes 2008, LNCS 5089, 258, 2008.
- [3] Gresse Von Wangenheim et al.; IEEE Software, **23**, 91, 2006.
- [4] Dragan Stankovic et al.; The Journal of System and Software, **86**, 1663, 2013.

주체105(2016)년 9월 5일 원고접수

A Method for Improving the Processes of the Small Software Development Units

Mun Il Nam, Ryang Ki Son and Kim Chol Song

We discuss about the problems arising in improving the processes of the small software development units and propose a simple but practical light-weight SPI method to provide a solution to them.

Key words: software development, software process, SPI