

우리에 맞는 Embedded Software 개발 환경에 대한 고민

Insung Park
Research Fellow (VP)
Software Platform Lab
LG Electronics

Introduction

Education

- 서울대학교 제어계측 공학과 학사
- Virginia Tech VLSI 전공 석사
- Purdue 병렬 처리 전공 박사

Work Experience

- **Microsoft 14 년 근무 ('00년~'14년 4월)**
 - Core OS Fundamentals 팀 (9년 6개월)
 - 성능/안정성 파트 개발 팀장
 - XP, Server 2003, Longhorn, Vista, Windows 7 개발 참여
 - Internet Explorer 팀 (4년 6개월)
 - HTML Controls, CSS, HTML Editing 파트 개발 팀장
 - IE9, IE10, IE11 개발 참여
- **LG 전자 Software Platform 연구소 입사 ('14년 4월~)**

가장 많이 듣는 질문

“왜 한국에 돌아오셨어요?”

“한국 SW 개발 환경은 (선진 SW 회사에 비해) 어떻게 다른가요?”

“(열악한 환경에서) 갑갑하지 않으세요?”



Context

~2000 Windows 95/98

품질 및 성능 이슈

1995~2003

2001 Windows XP

- Windows 3.x 기반의 App 환경을 WinNT 기반으로 변경
- 코드 품질, 성능, 안정성의 혁신적 개선

2003 Windows Server 2003

- 성능, 안전성, 보안, 서버 관리 기능 중점

한 Software 보안 강화 시스템 구축

- Security Development Lifecycle 구축
- Threat Modeling, 정적 분석 기법 및 툴 개발/강화
- 현재까지도 MS 는 Software 보안 프로세스의 선두



Context

— 1995-2003

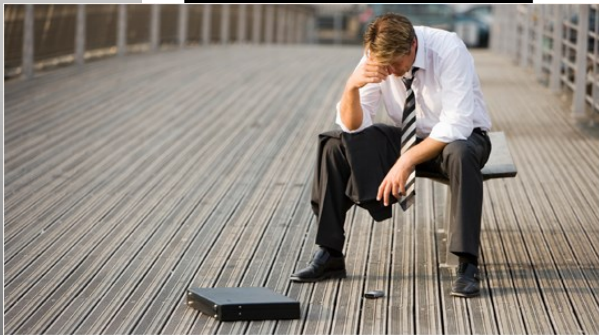
name Longhorn 개발 중단

- 비크 수는 계속 늘고,
- 기능 구현은 미뤄지고,
- 안정된 빌드가 드물고.
- 현 codebase 로는 제품 출시가 불가능하다는 결론에 도달
→ Back to Server 2003 codebase!
- XP 개발 때의 소프트웨어 공정의 실패
- MS 의 가장 뼈아픈 실패 중의 하나

2003-2005

Engineering System 필요

- 소프트웨어 공정 전문 인재 영입
- Engineering System Group 강화
- Goal: "Main branch can be released at anytime!"



Context

— 1995-2003

— 2003-2005

2005

Engineering System 확립

ure 관리 시스템

ec 검증 필수

스트 동반 필수

한 Code Review 시스템

kin 자동화

엄정 관리 (Template, Bug Jail 등)

한 Branch 관리

ality Gate (Performance, Reliability, Compatibility)

환경 개선

등 툴 강화

드 검색 용이

속적인 빌드 성능 개선

현 빌드 저장 (VM)

t coverage 접근 용이

— 1995-2003

— 2003-2005

— 2005

● 2005-Present

꾸준한 Engineering System 개선

- 지속적인 툴 및 프로세스 개선
- Agile Development 적용
 - 짧아지는 Release Cycle
- Telemetry 를 통한 사용자의 요구 적극 수용
- One Microsoft – One Codebase
- Open Source 수용 & 지원



Microsoft 의 기업 특성

- 대기업 : Big Momentum, Little Innovation
- 큰 실패를 여러 번 경험함
 - Google 의 성장 방치
 - Smart Phone 시장에서의 추락
 - Longhorn Reset
 - IE 의 추락, ...
- “최적화된 시스템은 없다”
 - 지속적인 프로세스 & 정책 개선
- 냉철한 실패 분석을 통해 다시는 실패하지 않도록 시스템 구축
 - 직원들이 믿고 따르도록 교육과 독려, 평가와 개선
- IT 산업에서 40년간 선두적 지위 유지

Google 의 기업 특성

- 정밀한 Software Engineering
- All Open Policy
 - 투명한 운영과 Communication
 - 세계 어디서나 모든 프로젝트 소스/서류 접근 가능
- Manager 는 인원 관리 및 HR 에만 집중
- 프로젝트 진행은 Technical Leader 이 주관
- 3 개월마다 프로젝트 Review
- 회사 내 자유로운 이동
 - 특별한 인터뷰 없이 조건이 맞으면 수시 이동
 - 성공 가능성이 높은 과제를 추려내는 기능

이상적인 Software 개발 환경

- 정답은 없다!
- Microsoft Windows
 - 세계에서 가장 큰 Software 프로젝트
 - 여러 번의 실패를 통해 지금의 시스템 구축
- 프로젝트 규모/인원/릴리즈 일정/특정 외부 요인에 따라 적합한 환경이 다름
- 무분별한 benchmarking 은 오히려 Overhead 만 야기

한국의 Embedded Software 개발 환경

- Hardware 개발에 따른 한계
 - 제품, 일정, 요구 사항의 잦은 변경
 - dogfooding 의 어려움
 - Software 변경 impact 에 대한 이해 부족
- 체계적인 프로세스 개선 여력 부족
 - “출시하느라 바쁘니까...”
 - 프로젝트 관리 Knowledgebase 미비
 - 특정 프로세스의 당위성에 대한 팀원들의 이해 구하는 노력 부족
- 조직 문화 특징
 - 명확한 역할 분담이 필요
 - 건설적인 토론과 합의 부족
 - 권한과 책임의 집중
 - 조직의 목표 > 프로젝트의 목표

“Software 기능 요구 사항이 빈약합니다.”

“모든 업무 요청이 Priority 0 로 옵니다.”

“관련 부서와의 협업이 힘들 때가 있습니다.”

“구성원이 창의성을 마음껏 발휘할 수 있었으면 좋겠어요”



LG전자의 Software Engineering

- Software Engineering 의 중요성을 알아가는 과정
 - 최근 3년간 급진적인 개선을 이룸
 - Android 협업, webOS 적용, 자동차 OEM 대응을 통해 선진 프로세스를 경험하고 내재화하고자 노력
 - SCM, 검증 자동화, 정적 분석 등에 많은 투자
 - Global 개발 협업 체계 구축 및 지속적 개선
 - 구성원이 아이디어를 가지고 직접 진행해 볼 수 있는 제도



- 최근 LG전자 Software 센터 중점 추진 사항
 - 전문 지식과 인력 확보
 - Program Manager, SDET, Architect 육성 프로그램
 - Coding 전문가 인증 및 활용 프로그램
 - Software 교육 및 역량 평가
 - Agile Development & Rapid Prototyping
 - 실시간 프로젝트 관리 체계 개선
 - 프로젝트 Postmortem 및 관리 Knowledge 축적
 - SW Platform 검증 Test Suite, Test Coverage 확대
 - 개발 및 검증 툴 증대
- “3 년 전에 비하면 (SW 개발 환경이) 엄청 좋아졌어요.”

LG전자의 Software Engineering



<Software Architect Course >



< Software Architect Workshop >



< Coding Expert Certification >



< SDET Certification >

Thank You