



사물인터넷 융합 기술과 오픈소스



August 6, 2015

경북대학교 IT대학
교수 강순주



발표자 이력

- KAIST 전산학과 석박사(SE Lab 1회)
- 1985~1995 : 한국원자력연구원 선임연구원,전산정보실 실장
- 1996 ~ 현재 : 경북대학교 IT 대학 교수
 - 과학위성3호 보조컴퓨터용 운영체제(KAIST SETRAC)
 - 무인 헬기 자율 비행을 위한 소프트웨어구조 연구(과학재단)
 - 홈네트웍 게이트웨이 셋탑 공통 소프트웨어 구조(ETRI,린나이 코리아)
 - 임베디드 소프트웨어 개발 환경 연구센터 (ITRC(센터장 선문대 임기욱))
 - 자원제약형 임베디드 시스템을 위한 RTOS(Ubinos) 개발 (Open source)
 - USN형 화재방범 예방 시스템(uFIPI) → (주) ART System 기술 이전 → 서울시 산하 지하상가 전역 설치 운영중
 - 자율군집소프트웨어플랫폼 연구 센터장(2011 ~ 2016, 정부출연금 100억)
- 산학협력대상(2015) – 경북대 총장
- 우수성과 상(2014) – 미래부 장관
- 국가연구개발우수성과 100선(2014) – 미래부/산자부 장관상

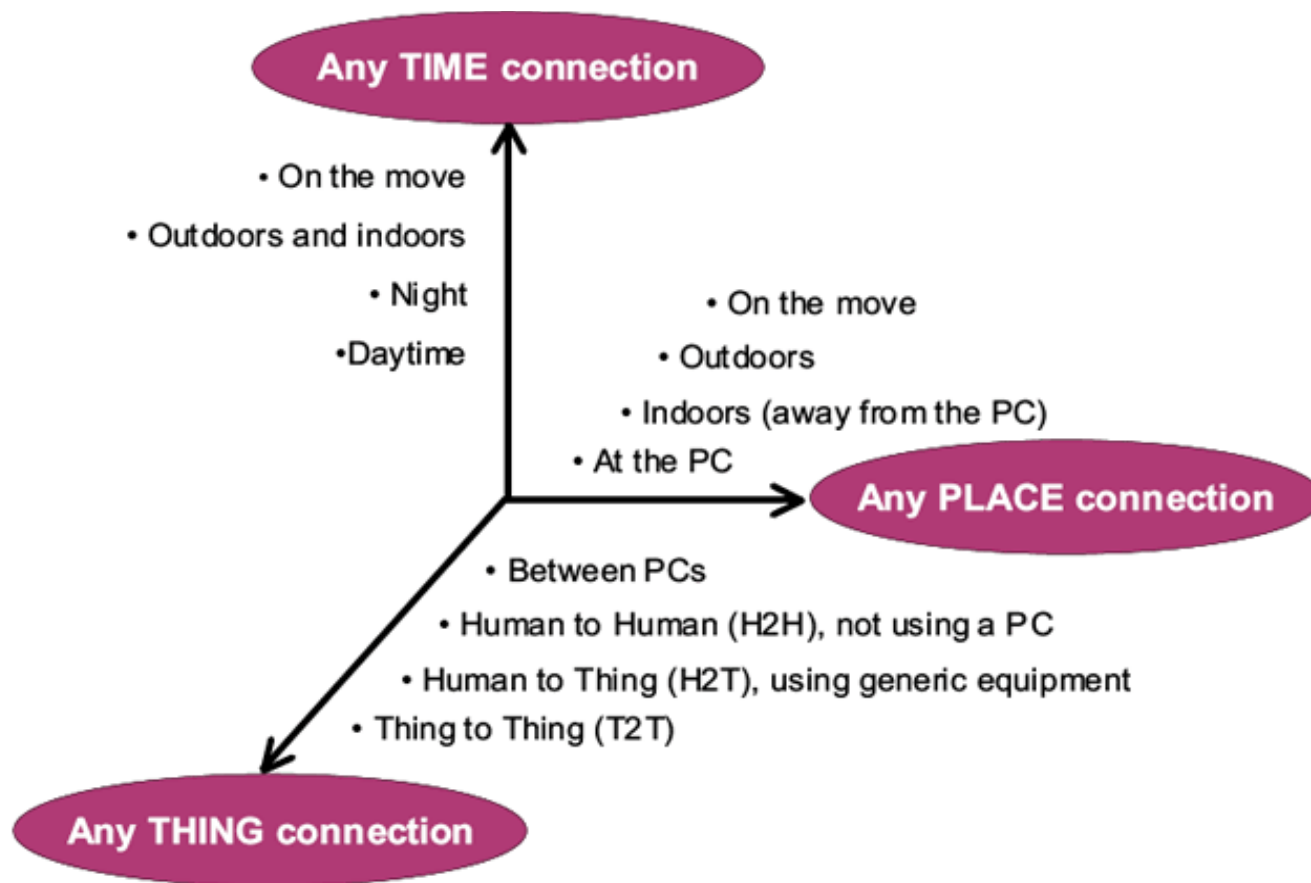


Index

- 01 IoT 개요 와 문제점
- 02 IoT 기술 개발에서 Open Source 활용
- 03 소프트웨어 개발 공정(SDLC) 과 CICD
- 04 CICD 실현을 위한
오픈소스 기반 도구구성
- 05 KNU rtlab ubinos 개발 사례
- 06 Subversion, Trac, Doxygen 설치 및
사용 시범
- 07 IoT 환경에서의 Open Source Projects
- 08 결론



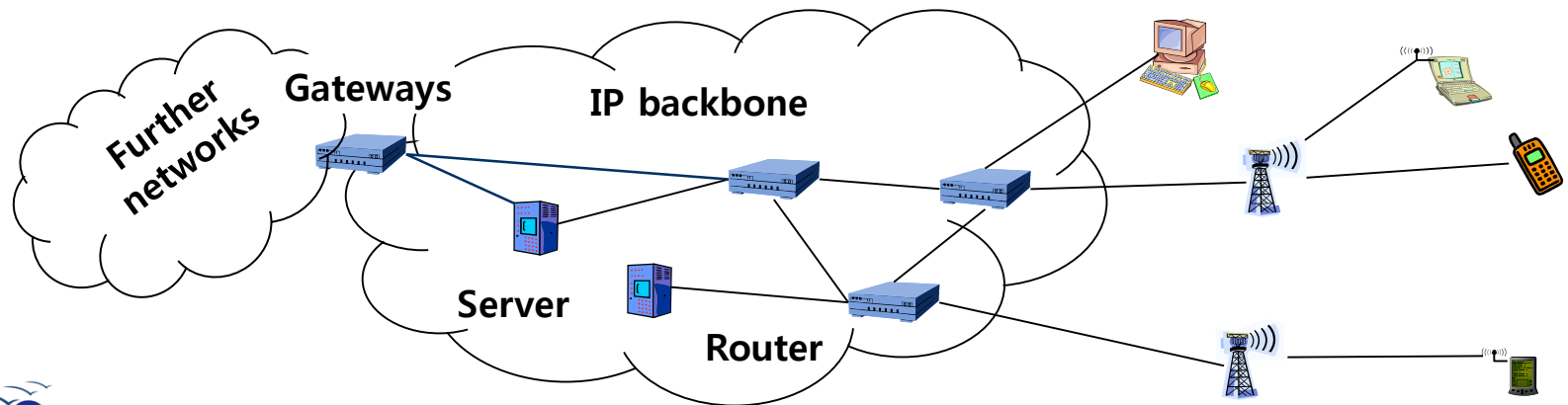
(인용: Helen Gill, Ph.D. CISE/CNS NSF presented in the Symposium of Safe & Secure Software and Systems, June 15,2010)

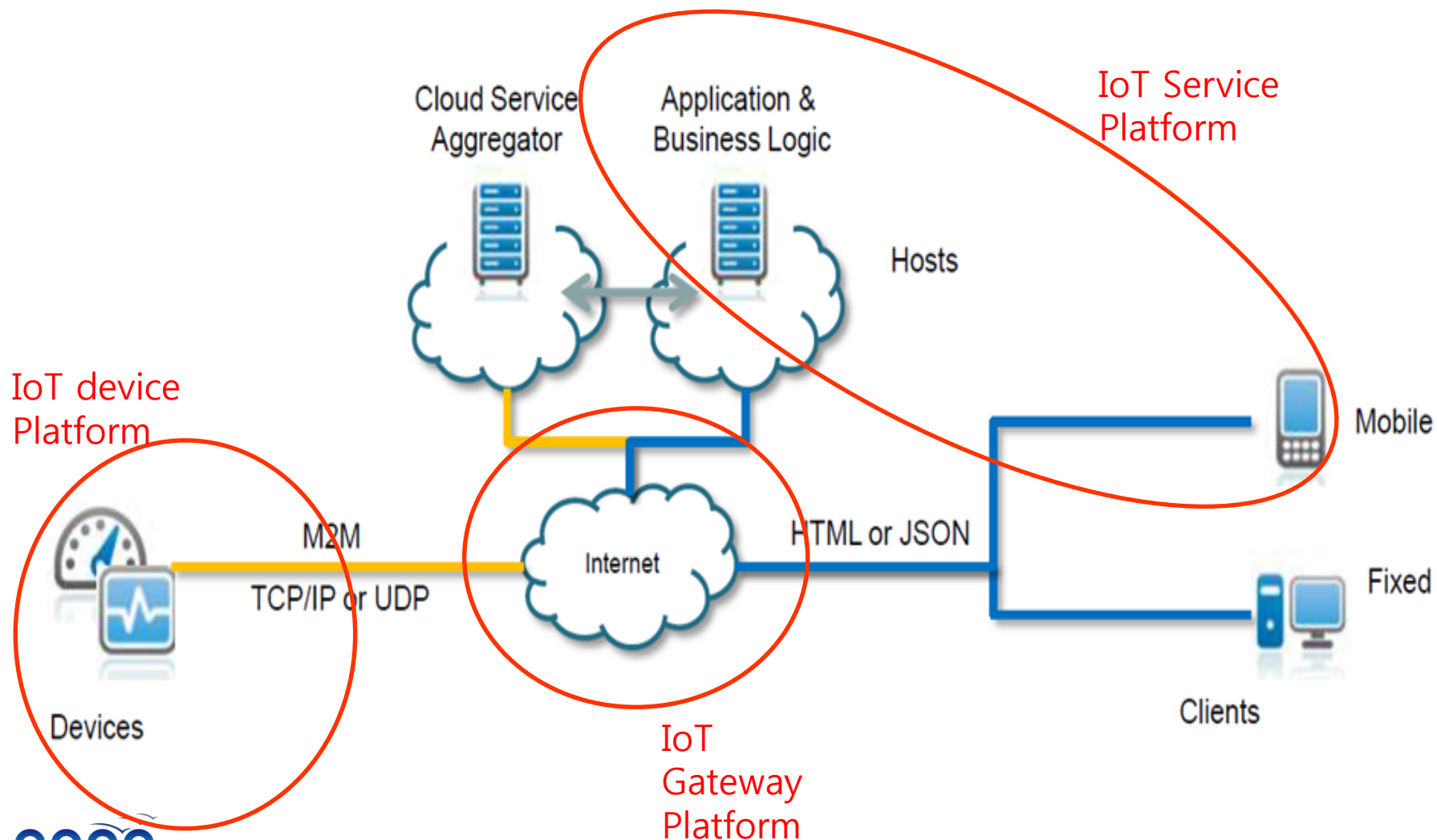


Source: ITU adapted from Nomura Research Institute

- How to build the service infra-structure ?
- Is it materialized the sensor/actuator interaction in real-time ?
→ Guarantee the determinism and solve the security issues ?
- Why and how to messaging among mobile and stationary IoT ?
- The Big Data technology increase the quality of human life ?
- Applying Open Source ?

- Typical wireless network are based on infrastructure
 - E.g., GSM, UMTS, WLAN, ...
 - Base stations connected to a wired backbone network
 - Mobile entities communicate wirelessly to these base stations
 - Traffic between different mobile entities is relayed by base stations and wired backbone
 - Mobility is supported by switching from one base station to another
 - Backbone infrastructure required for administrative tasks





- 기존의 인터넷망(공중망 포함)은 Things을 제어하는 용도가 아님
→ sensing/actuation in real-time (X)
- IoT 서비스 자체가 지역적인데도 불구하고 모든 데이터를 중앙에 집중 저장 관리 분석함
→ 엄청난 에너지 낭비
- 엄청난 데이터양(Everyday around 20 quintillion (10^{18}) bytes of data are produced에 비해 활용가치가 높지 않음
- 통신사업자(Big Brothers)가 사업권을 독점하여 새로운 비즈니스 틈새가 없음. 지극히 개인적인 데이터(생체 신호, 가정제품 활용 정보, ...)를 글로벌 통신 사업자가 독점

Big Data
≈ Big Brothers



Forget 'the Cloud'; 'the Fog' Is Tech's Future

I'm as big a believer in the transformational power of cloud computing as anyone you'll meet. Smartphones, which are constantly seeking and retrieving data, don't make sense without the cloud, and any business that isn't racing to push its data and software into someone else's data center is, in my view, setting itself up for disruption by a competitor who is.



Brian Ajhar

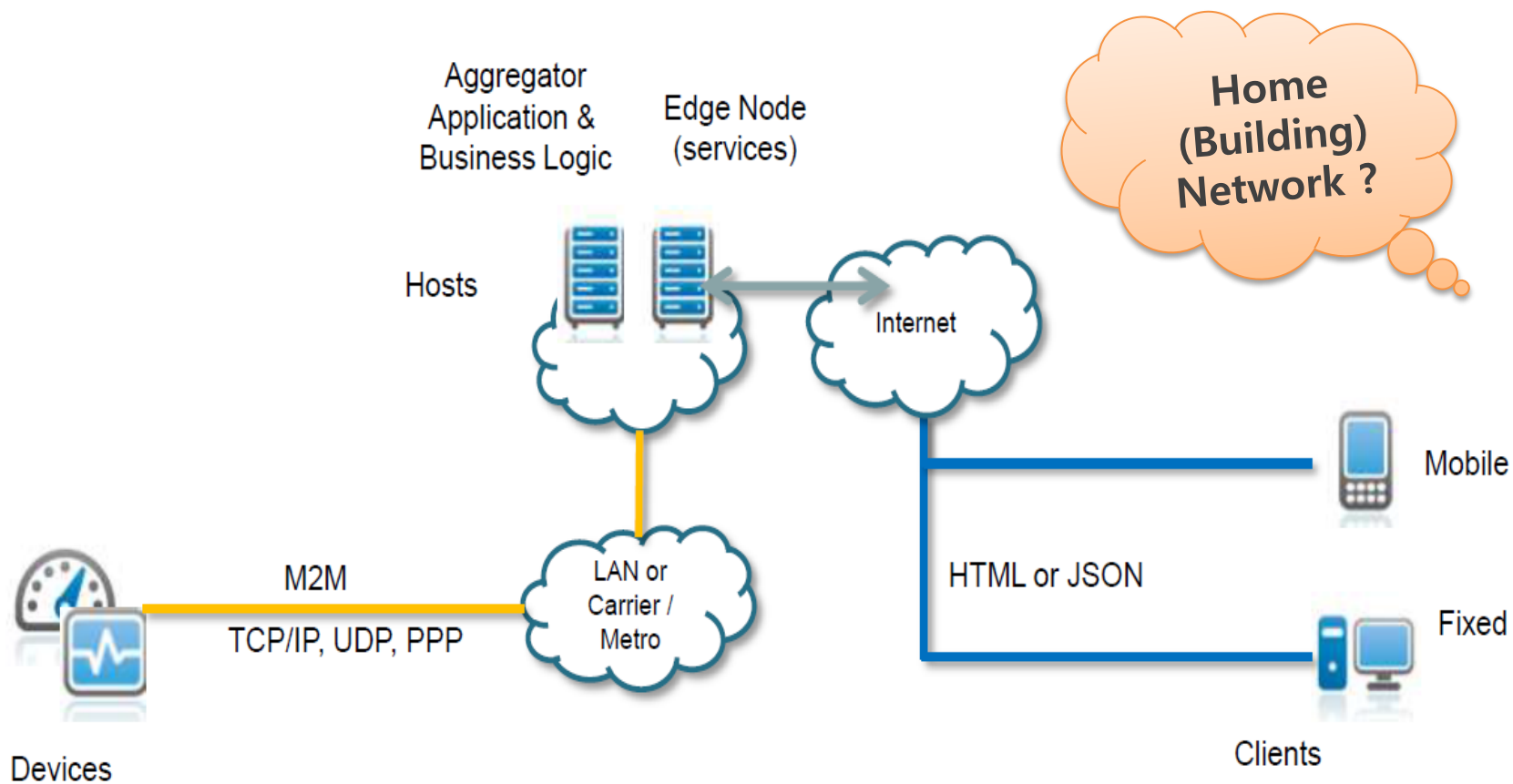
But cloud advocates are fond of declaring that 100% of computing will someday reside in the cloud. And many companies are in business to sell you on that notion.

Here's the reality: Getting data into and out of the cloud is harder than most engineers, or at least their managers, often are willing to admit.

The problem is bandwidth. If you're a company simply seeking to save the cost and headache of storing data yourself, the cloud is great as long as all you need to do is transfer data back and forth via high-speed wiring.

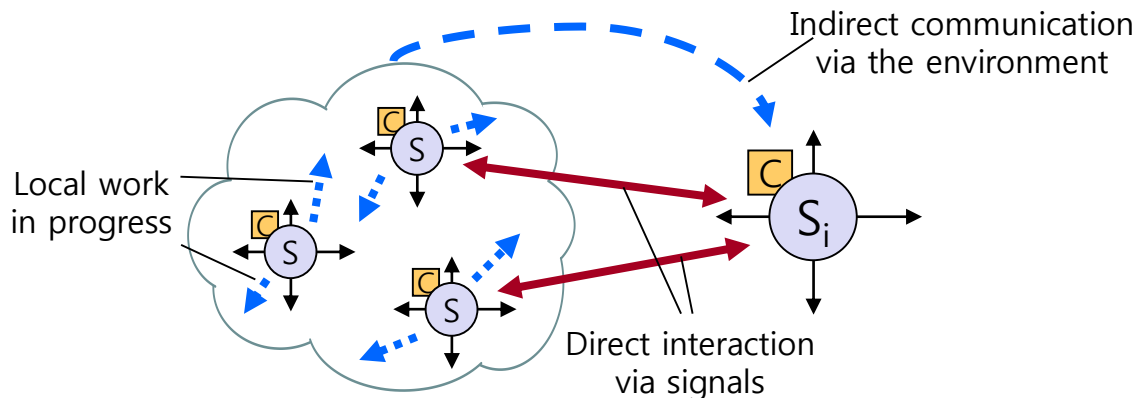
But in the world of mass connectivity—in which people need to get information on an array of mobile devices—bandwidth is pretty slow. Any business that sends data to mobile devices, be it airline reservation systems for consumers or business data for a mobile sales force, grapples with the limitations of wireless networks. Overall, according to the World Economic Forum, the U.S. ranks 35th in the world in terms of bandwidth per user.

**Write to Christopher Mims at
christopher.mims@wsj.com**





- Introducing Among Individuals and with the Environment
 - Direct communication among neighboring systems (gossiping)
 - Indirect communication via the environment (stigmergy)
 - Interaction with (stimulation by) the neighbors and environment
 - Identifying location of itself and neighbors
 - Clustering and collaboration
 - Delay Tolerant Communication
 - **Opportunistic service**
(activated when the time, the location, the users just matched)



02 IoT와 Open Source 공합(?)

- IoT 핵심 요소 기술을 Open Source 활용
 - 시스템 소프트웨어(RTOS, GNU, DBMS, etc)
 - 통신스택(Btstack ,Middleware ,etc)
 - 위험하고 상용제품 개발에 활용하면 안됨(각종 라이선스)
- 개발환경에서 Open Source 활용
 - 상용 환경보다 우수함(Eclipse, SVN, GitHub, GNU, etc.)
 - Global Standard이기 때문에 상용을 사용하더라도 이 분야의 추이는 파악해야함



공짜로 쓰자 (X)

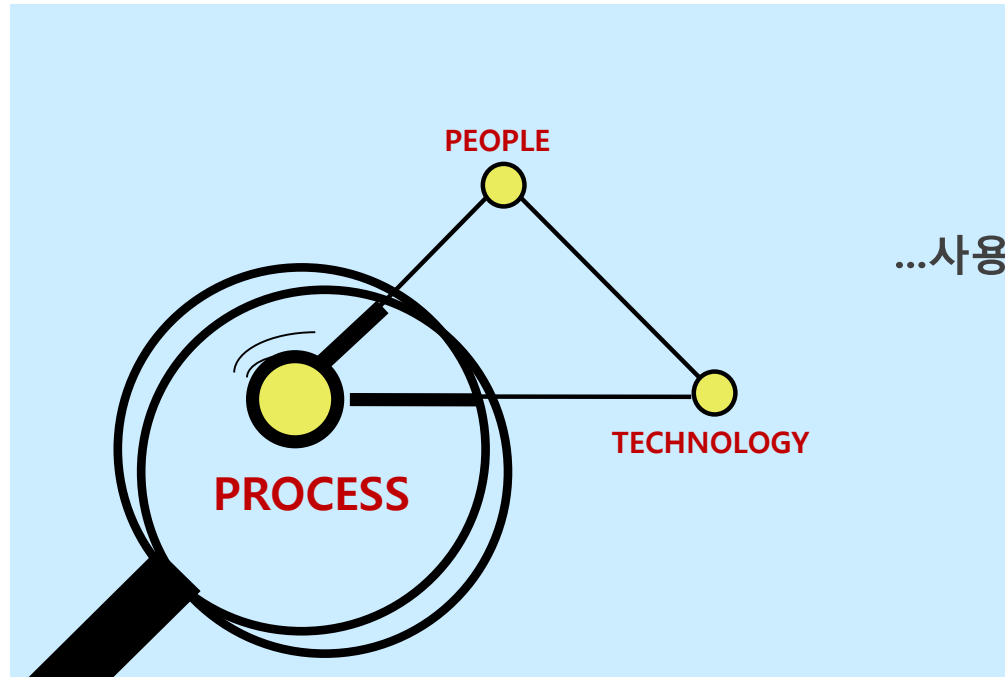


함께 개발하자(O)

- 소프트웨어 품질 보장할 능력 (?)
 - 글로벌 표준에 맞는 도구 및 환경 사용법 숙지 여부
 - 최신 개발환경 변화에 따라갈 기본기
- Software Engineering for Embedded Systems

- 미국 카메니멜론대 SEI 연구소 개발 보급한 소프트웨어 개발 공정의 성숙도
- 기업의 Critical SW 개발 능력에 대한 척도
- 5 단계 성숙도 레벨을 규정하여 특정 기업들에게 심사를 통하여 자격증 부여

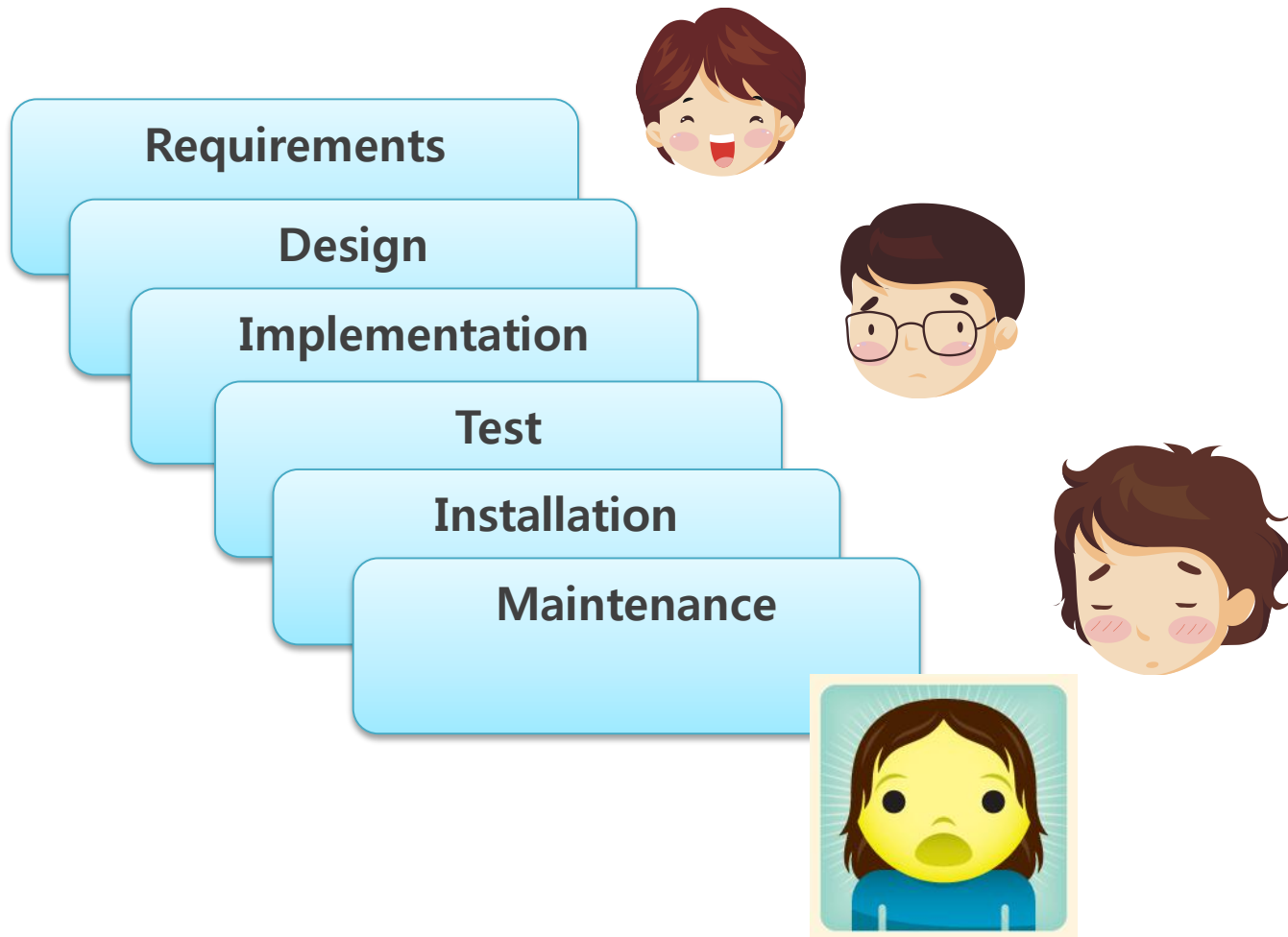
사람 ???



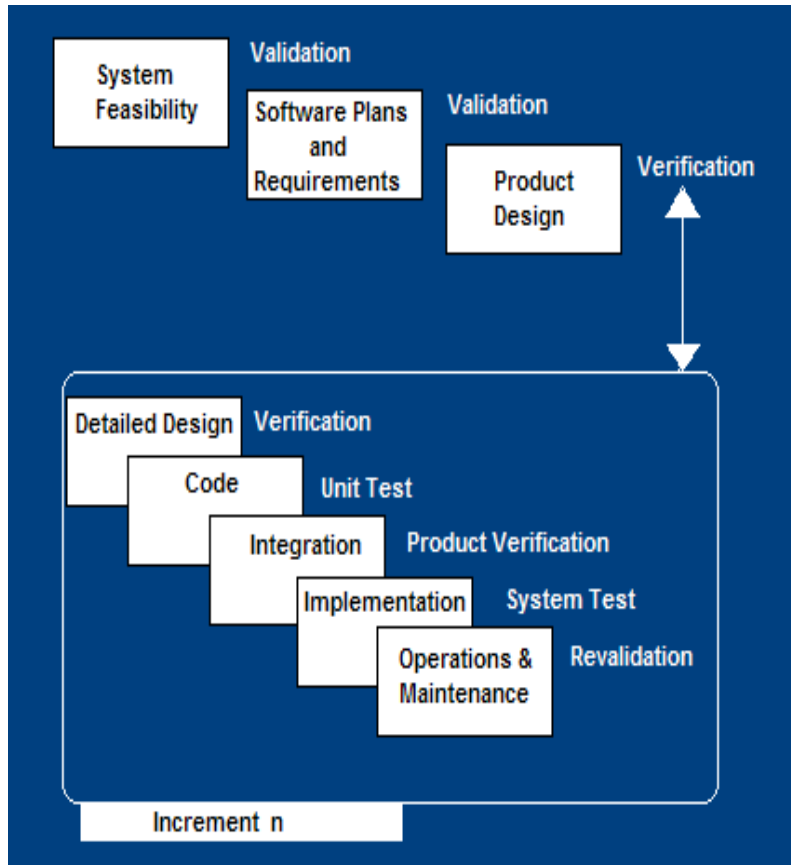
품질을 결정하는 요소는 해당 기업이 사용하고 있는 공정의 합리성...

03 소프트웨어 개발 공정 (Software Development Life Cycle Model)

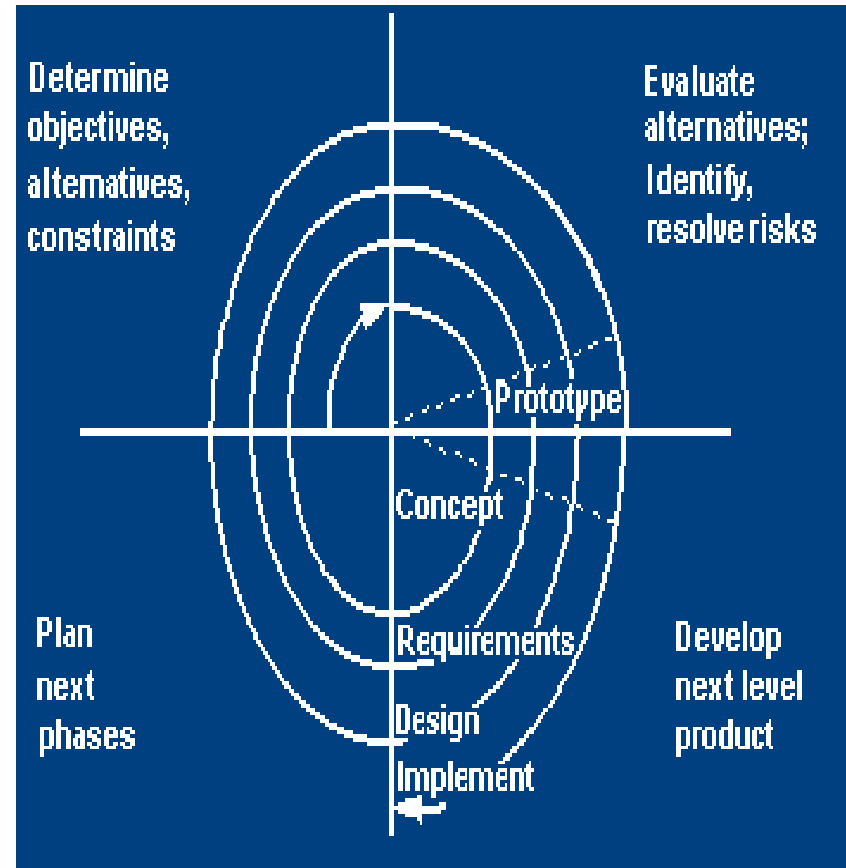
- 소프트웨어 개발 프로젝트에서 개발 세부 절차를 규정하고 각 절차(단계)별로 해야 할 작업들을 규정한 체계(공정)

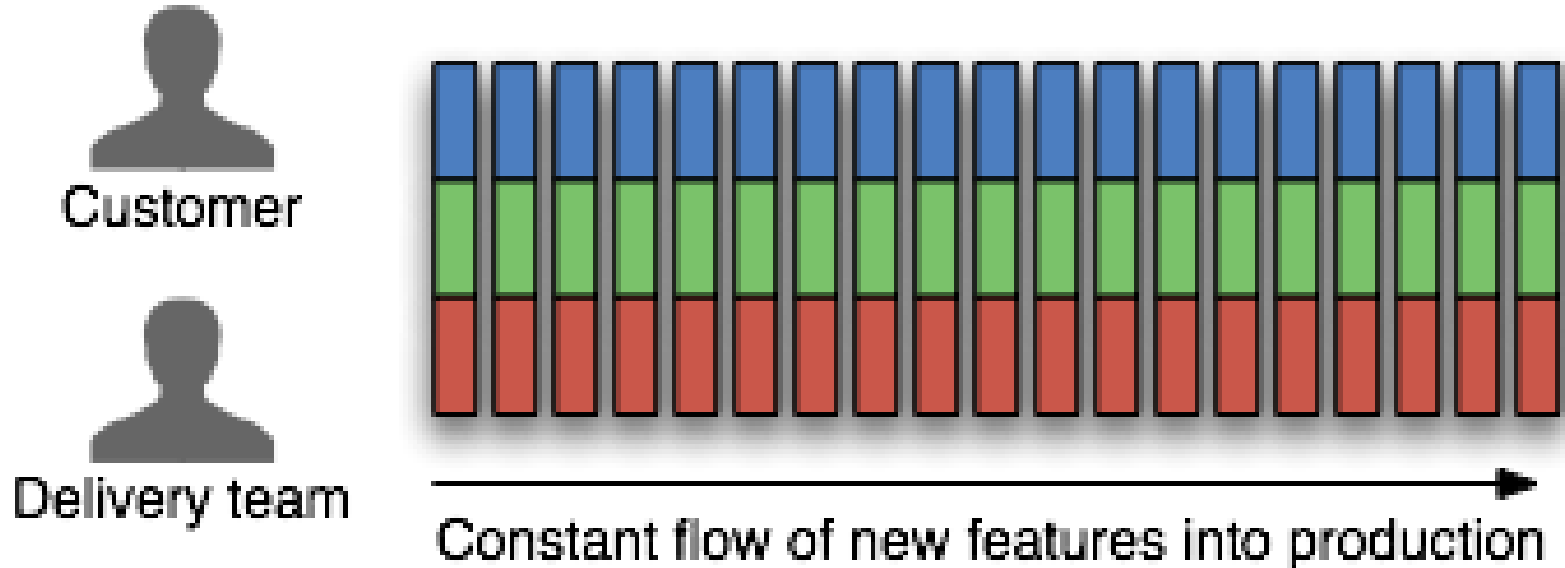


점증(Incremental) 모델



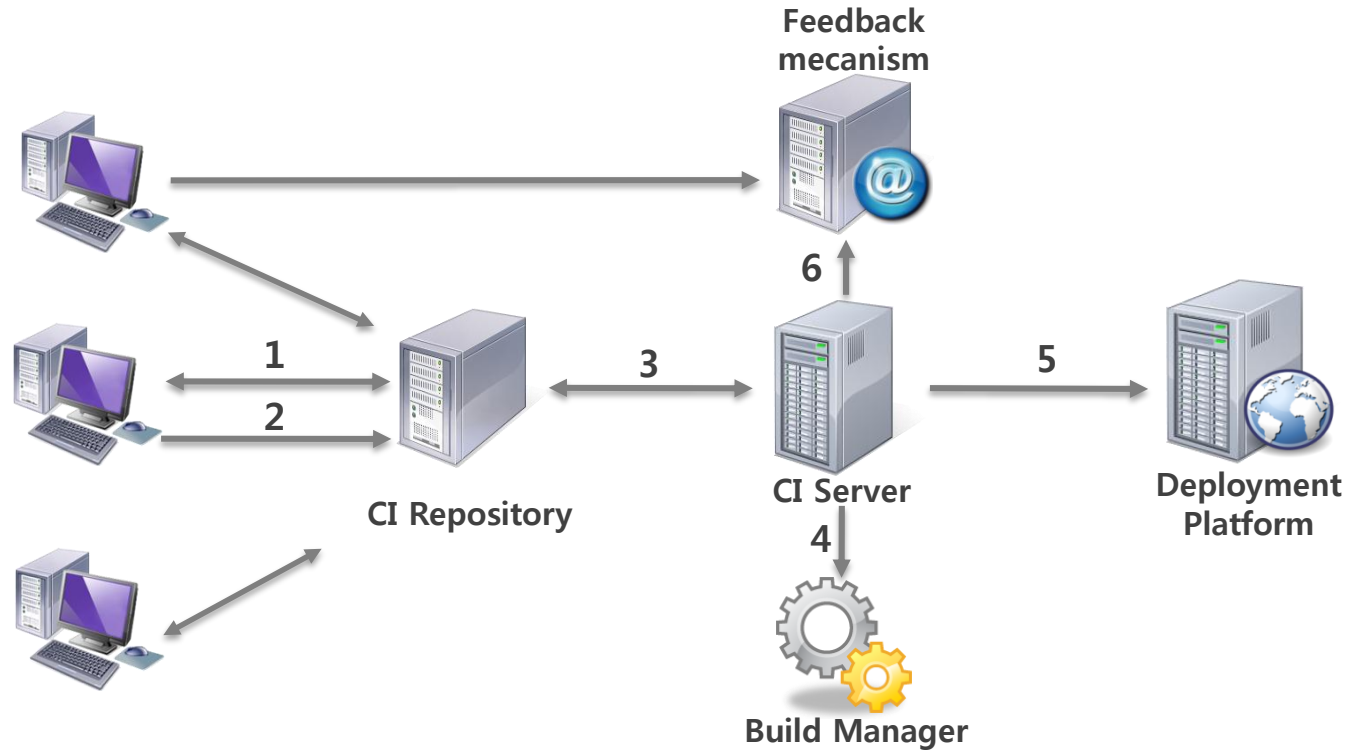
나선(Spiral) 모델



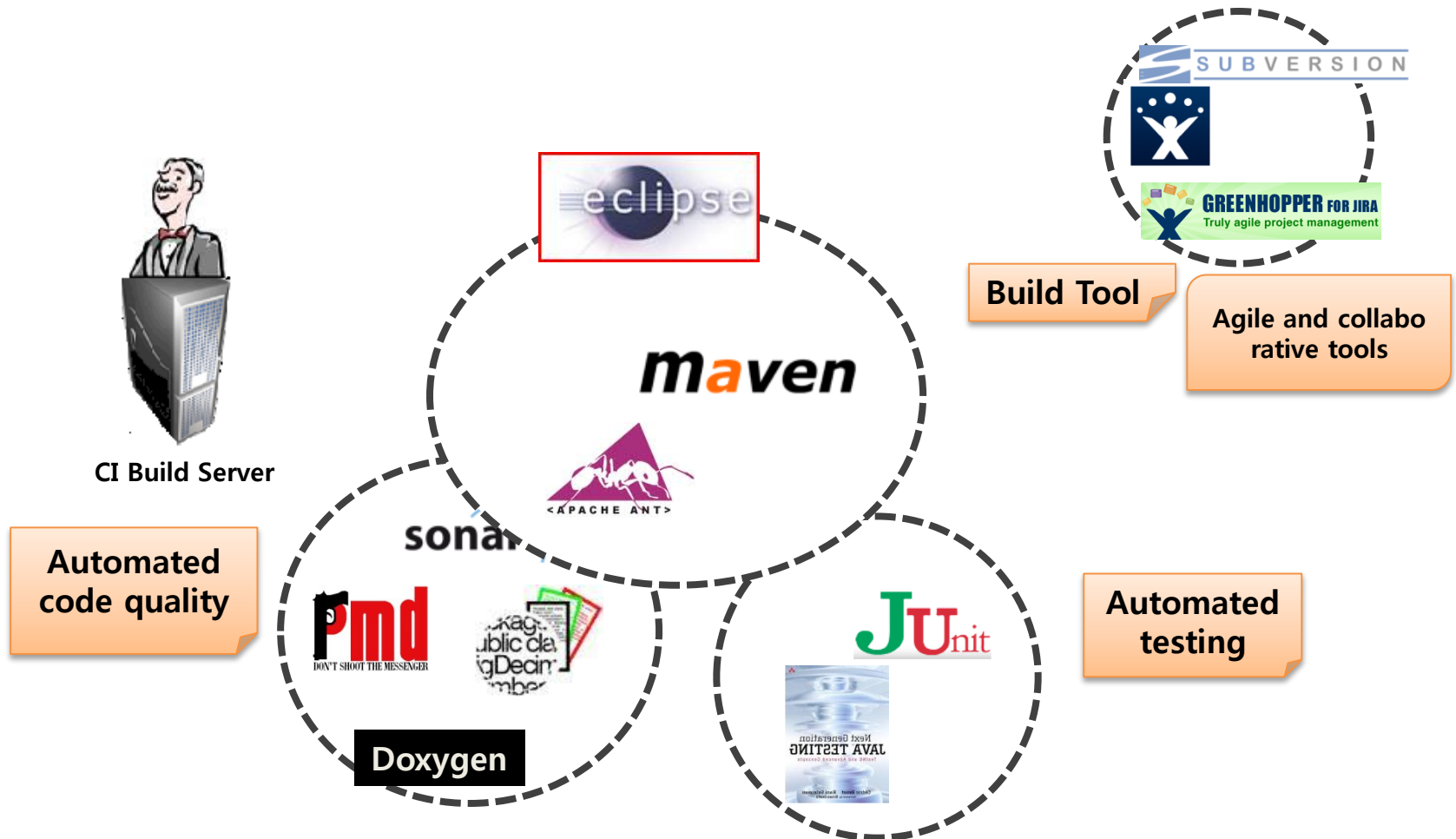


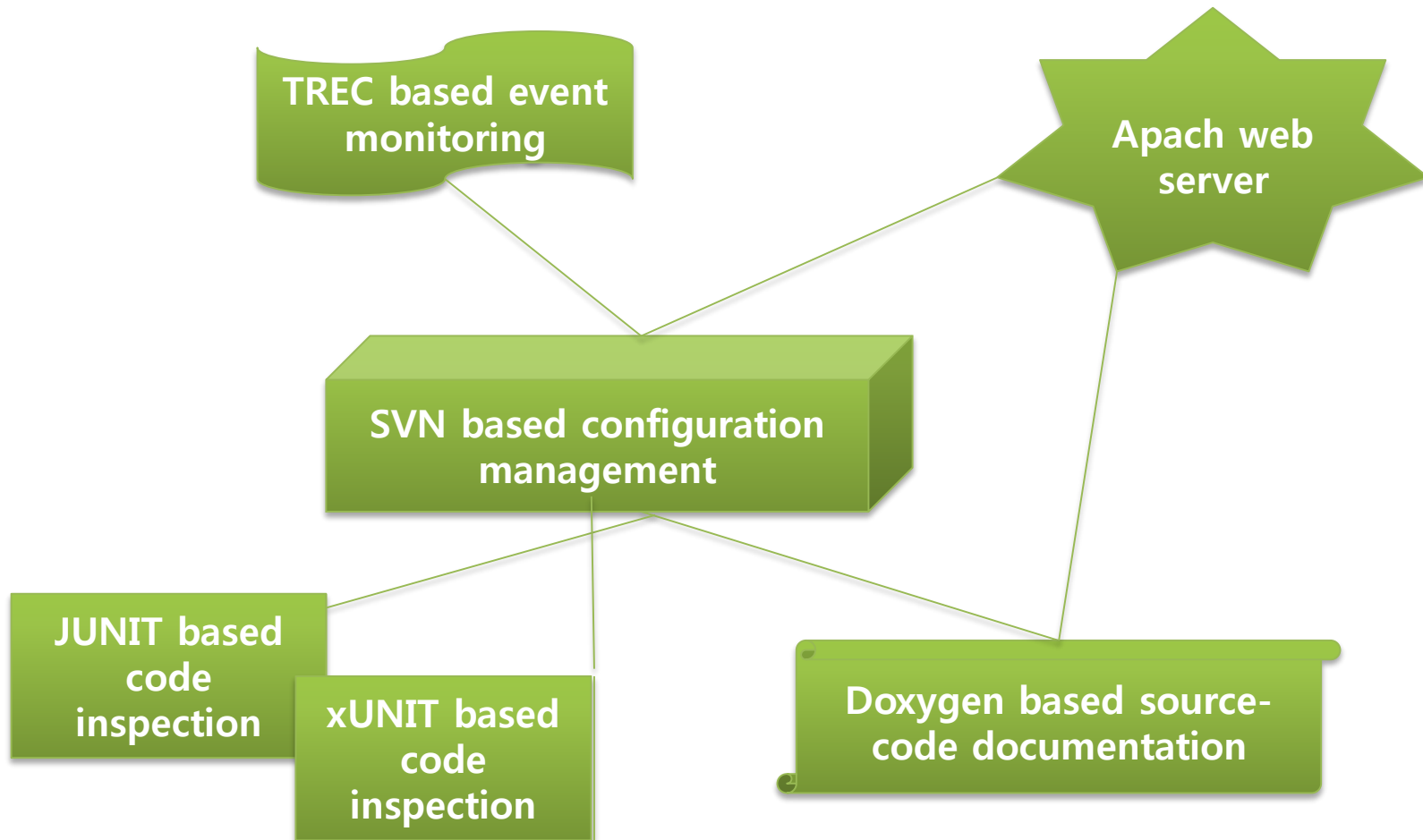
- 개발중인 소프트웨어가 언제든지 상품화 가능
- 출시 버전은 바로 영업부서와 바로 연동
- Continuous Integration 확장된 개념

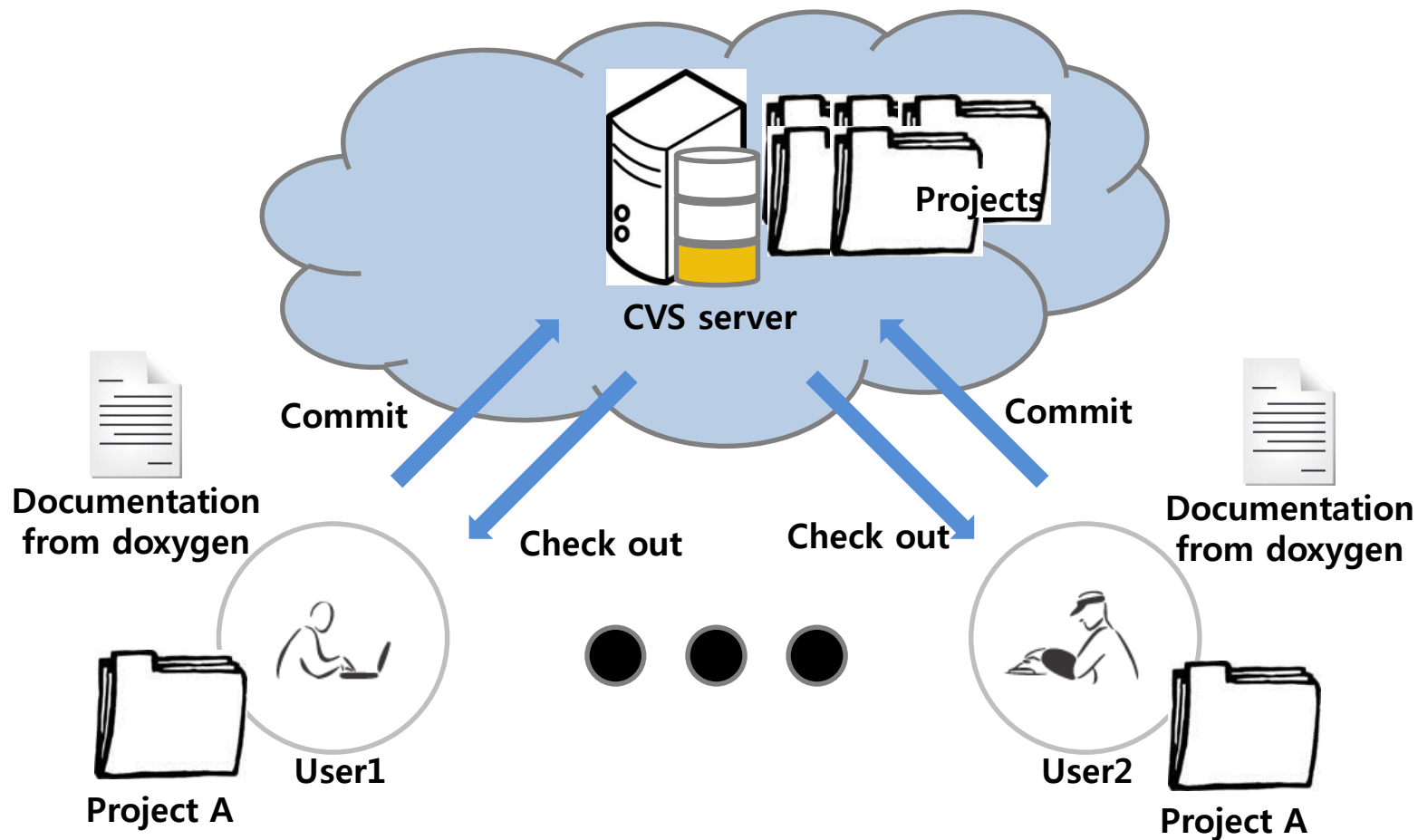
- 신속하게 소프트웨어 결과물 산출을 위해 가능한 많은 개발 도구 활용
- 개발 중 변경 사항이 바로 결과물에 반영
- 관리자,개발자,고객이 동시에 결과물(상품) 공유 가능
- 관리자,개발자 모두 현재 프로젝트 진행 과정 모니터링 가능



6.Feedback: feedback mechanism like email notification and groupware.



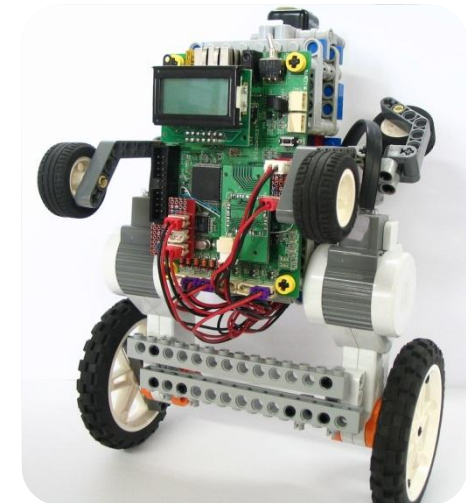
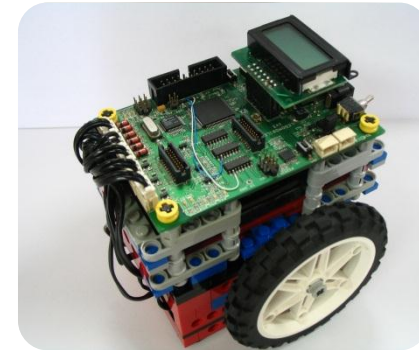
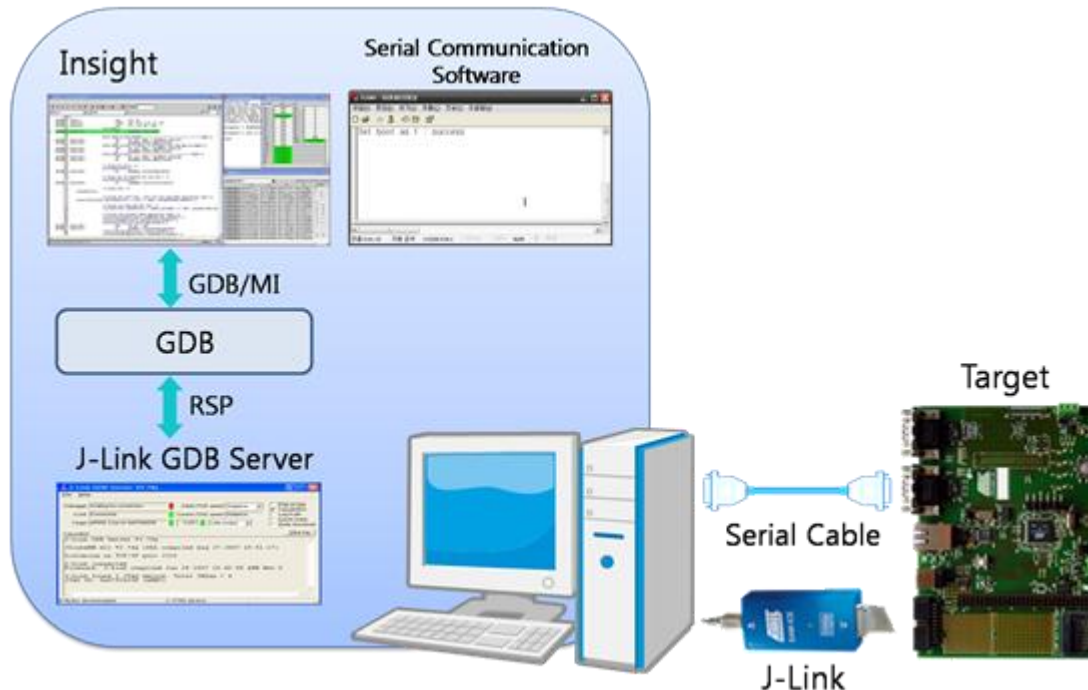


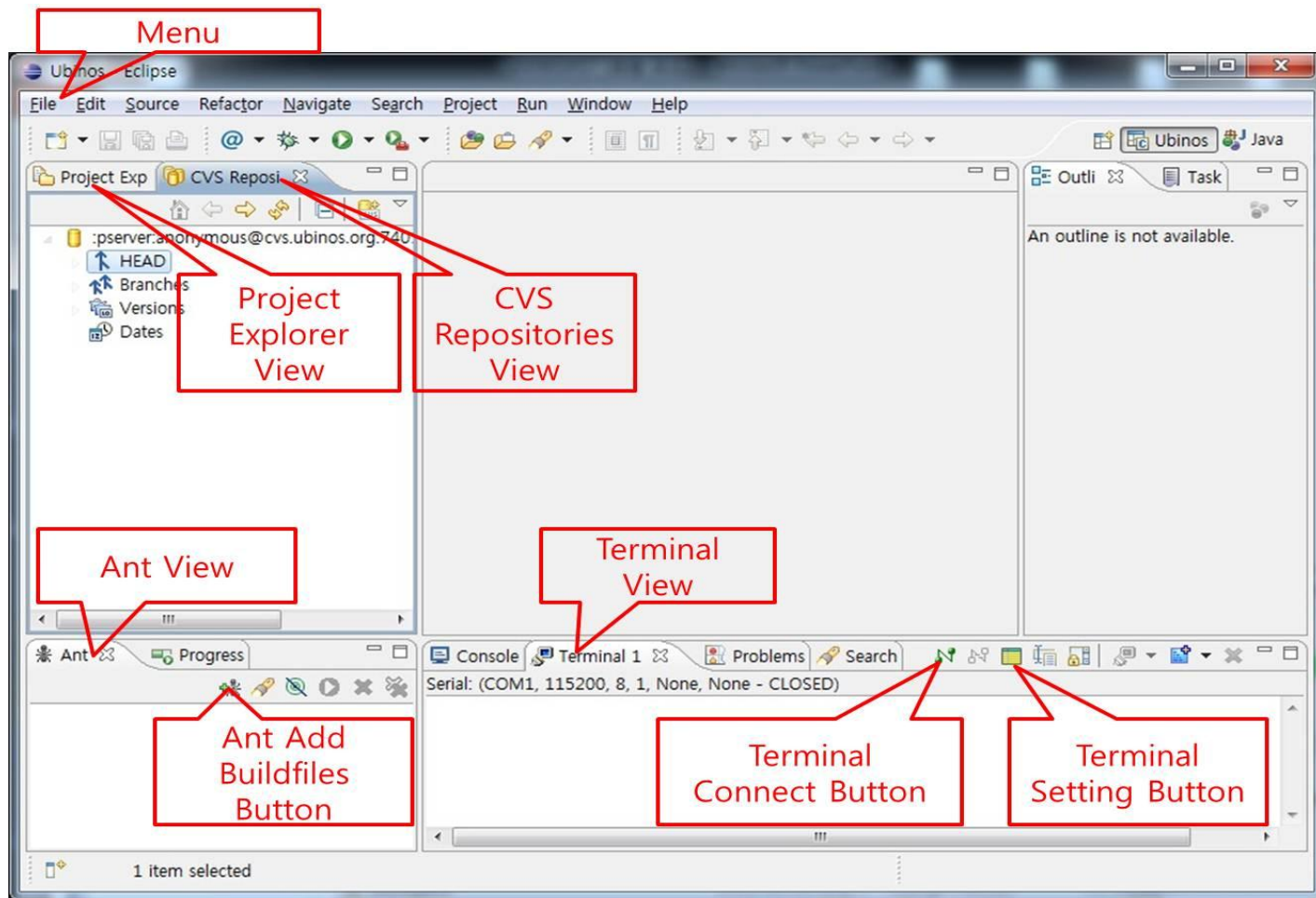


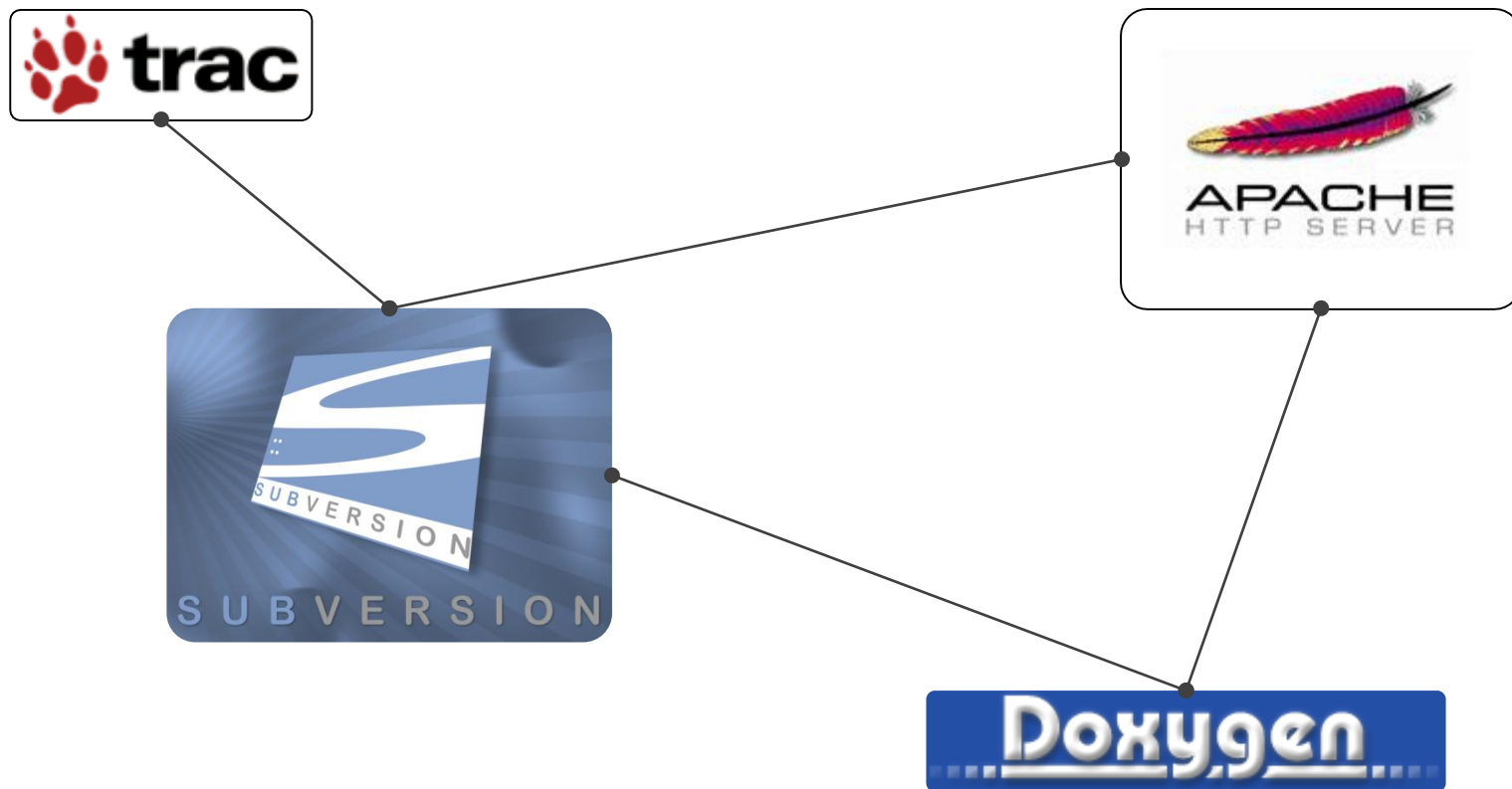
- Eclipse IDE 3.4.2
- GNU Toolchain
 - GCC 4.3.2 compiler
 - Binutil 2.19 Assembler linker
 - GDB & Insight 6.8 Debugging tools
- Ant 1.7.1
- Groovy 1.5.6(Gant)
- Doxygen 1.5.6
- Python 2.5
- KNU RTLAB RTOS & Library
 - Ubinos
 - Lego API
 - CAN, Zigbee API
 - lwIP – open source TCP/IP stack
 - Etc...

03 Cross-Compile Environment

- Host computer(JDK, Ubitools, Hyper Terminal)
- Serial Cable, J-Link
- Target System(ESPS Mobile board)





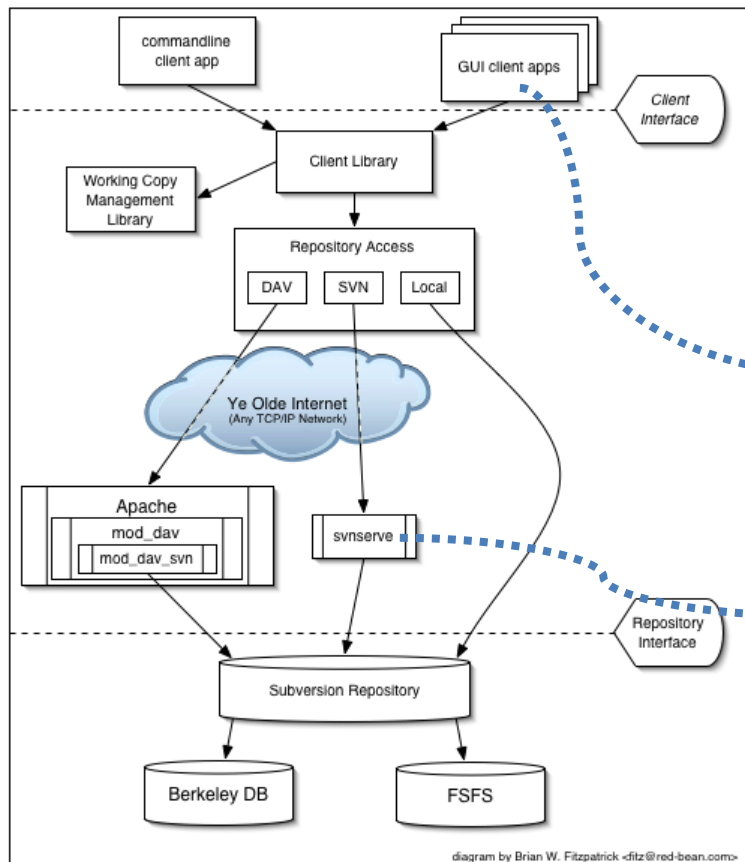


- 아파치 서버 및 메일 서버 설치
- Subversion 서버 설치
- Trac 서버 및 필요한 plugin 설치
- 아파치 + Subversion + Trac 연동 및 시스템 구성
- 사용자 아이디, 패스워드 생성
- 사용자 권한 할당

- 프로젝트 계획 및 로드맵 생성
- 로드맵 모니터링 및 관리
- 태스크 생성 및 개발자 배정
- 태스크 작업 상황 확인
- 버전 확인
- 개발자 재배정

- 할당된 태스크 accept
- SVN 서버에 프로젝트 공유
- 소스 수정 및 Commit
- 버그 리포트 및 defect ticket 생성
- 기능 추가 및 enhancement ticket 생성
- 태스크 완료 및 버전 fix

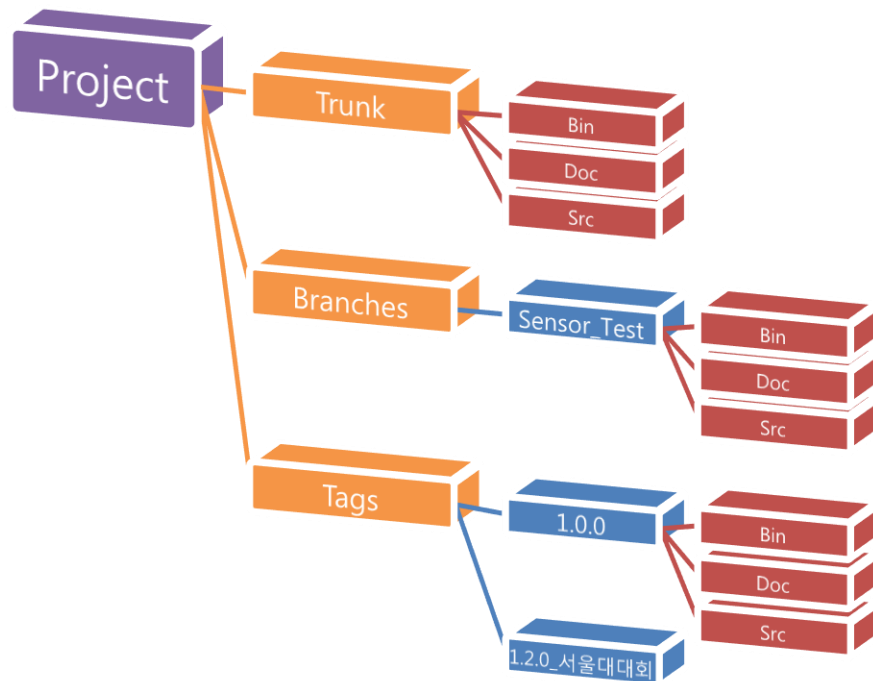
- Server
 - Subversion(<http://www.subversion.org>)
 - 커맨드 라인 기반의 클라이언트와 서버 프로그램
 - SVNserve Manager
(<http://www.pyrasis.com/.../SVNserveManager>)
 - Svnserve 명령어를 자동으로 실행시켜 편리하게 함
- Client
- Standalone → TortoiseSVN(<http://tortoisesvn.tigris.org>)
- Eclipse IDE plugin → Subclipse
- VS , Xcode plugin...



- 서버와 클라이언트로 구성되며 서로간에는 네트워크나 로컬로 접근한다.

- SVN Client (ex. Tortoise, IDE plugin)

- SVNserve Manager



- Trunk
 - 프로젝트의 중심이 되는 폴더
 - 실제 개발 진행은 이곳에서 진행
- Branches
 - Trunk에서 뻗어나온 또 하나의 프로젝트로서 메인 프로젝트에서 분기하여 따로 개발을 진행
- Tags
 - 버전 업이나 특정 시점에서 프로젝트 전체를 복사하여 보관하는 곳

- TRAC

- 기존의 cvs를 통한 오픈 소스 기반 형상관리 시스템의 단점을 개선한 subversion을 바탕으로 Wiki 와 이슈 트래킹 시스템을 추가해 프로젝트 관리를 가능하게 한 소프트웨어 형상관리+프로젝트 관리 툴
- Subversion이 오픈 소스 지만 각종 플러그인과 오픈 소스 개발자들의 노력으로 GUI환경에서도 편리하게 사용할 수 있도록 tortoise SVN이 나 IDE에서의 사용을 지원하기 위해 subclipse등의 활용이 가능해지므로 환경에 구애 받지 않고 사용이 가능한 장점을 흡수
- Wiki시스템의 실시간 정보교환의 장점과 이슈 트래킹을 통한 유지보수관리의 뛰어남이 TRAC의 최고 강점
- SPIC 통합저장소가 갖추어야 할 여러 요건들을 분석함에 있어서 가장 적합하다고 생각되어 우선적으로 벤치마킹 함

• View Ticket

- 마일스톤 이나 버전, 수정자 등을 통해 최근에 발생한 이슈들을 정렬해서 보여줌

Available Reports

This is a list of reports available.

Report	Title
{1}	Active Tickets
{2}	Active Tickets by Version
{3}	All Tickets by Milestone
{4}	Assigned, Active Tickets by Owner
{5}	Assigned, Active Tickets by Owner (Full Description)
{6}	All Tickets By Milestone (Including closed)
{7}	My Tickets
{8}	Active Tickets, Mine first

n (52 matches)

s by priority, while grouping results by version.

and reporter are included as hidden fields for useful RSS export.

		Component	Version	Type	Owner
#233	프론트로그 지원	Core		defect	papa
#71	rss 생성시 플래쉬, embed 처리	Plugins		enhancement	papa
		User Interfaces		enhancement	egoir
		Libraries		enhancement	?clay
		User Interfaces		defect	egoir

1.0

Ticket	Summary	Component	Version	Type
#45	EAF PageMaster의 스크립트 에러	Components	1.0	defect
#104	리스트를 볼때 특정 조건으로 검색시 페이지때문에 아무것도 나오지 않는 경우	User Interfaces	1.0	defect
#130	페이지 로딩 전 다중업로드를 사용 시 경고 메시지 발생	Components	1.0	defect
#138	블로그 도메인 변경	API	1.0	defect
#140	이미지 파일 누락	API	1.0	defect
#142	파콧에서 업로드 오류	API	1.0	defect
#149	첨부한 파일 열기 불가	User Interfaces	1.0	defect
#97	XMLStruct 개선	Components	1.0	enhancement
#105	트랙백 제목 영역에 대하여 링크를 알 수 없음	User Interfaces	1.0	enhancement
#202	이후에 추가되는 치환자를 위하여 미처리 치환자 제거 기능 추가	Core	1.0	enhancement
#44	블로그 화면에서 수정 시도시 임시저장본 체크 안함	Components	1.0	defect
#119	댓글에 치환자가 적용됨.	User Interfaces	1.0	defect
#191	트랙백 로그에서 글로 향하는 링크가 없음	User Interfaces	1.0	enhancement

03 TRAC을 이용한 실제 프로젝트 진행 사례(6)

• Search

- 검색 키워드를 통해 이슈를 검색

Search

☒ Tickets ☒ Changesets ☒ Wiki

Search results (1 - 10 of 92)

[bugs/1.1: == 알림 == ...](#)

... 로 로그인 하면 잘 되나, 이 상태에서 왼쪽의 카테고리를 선택하면 URL 주소가 http://jigi.dothome.co.kr로 바뀌면서 로그인이 풀려버림. *
[http://www.tattertools.com/ko/bbs/view.php?id=qna&page=1&sn1=&divpage=2&sn=off&ss=on&sc=on&select_arrange=headnum&desc=as ...
By inureyes — 12/20/06 20:13:49

[\[2680\]: * 비공개 카테고리 인터페이스 홀딩.](#)

* 비공개 카테고리 인터페이스 홀딩.
By inureyes — 12/15/06 16:23:33

[\[2669\]: * 상위 카테고리를 클릭할 때 비공개 카테고리가 ...](#)

* 상위 카테고리를 클릭할 때 비공개 카테고리가 드러나는 버그 수정
By inureyes — 12/15/06 00:44:25

[\[2668\]: * 페이지장하는 곳에서 비공개 카테고리에 속한 글 노출 ...](#)

* 페이지장하는 곳에서 비공개 카테고리에 속한 글 노출 차단.
By inureyes — 12/15/06 00:28:54

[\[2667\]: * 글의 경우 해당 카테고리가 비공개이면 출력하지 ...](#)

* 글의 경우 해당 카테고리가 비공개이면 출력하지 않도록 수정.
By inureyes — 12/15/06 00:23:47

알림

- 현재 태터툴즈 1.1에서 발견된 버그들의 처리상황 목록입니다.
- 이 목록에 포함되지 않은 문제점이나 의견을 포럼 게시판에 남겨주시면 최대한 빠른 시간 안에 처리하도록 하겠습니다.
- 태터툴즈는 사용자의 참여로 만들어지는 오픈소스 프로젝트입니다. 여러분의 참여를 기다리고 있습니다.

신규 문제점 (검토중 포함)

- Autostart 문제
 - ⇨ <http://www.tattertools.com/ko/bbs/view.php?id=qna&...desc=asc&no=6923> (by ⇨ duccobi)
 - WMP는 랜창인데 QuickTime이 문제이려나
- BR 증발
 - ⇨ <http://www.tattertools.com/ko/bbs/view.php?id=qna&...desc=asc&no=6918> (by Claire)
 - 엔터를 싣고 있음.
- URL 주소에 따라 로그인이 풀리는 현상.
 - \$service['domain'] = 'jigi.net'로 설정된 상태에서 www.jigi.net로 로그인 하면 잘 되나, 이 상태에서 왼쪽의 카테고리를 선택;
<http://jigi.dothome.co.kr>로 바뀌면서 로그인이 풀려버림.
 - ⇨ <http://www.tattertools.com/ko/bbs/view.php?id=qna&...desc=asc&no=6995> (by ⇨ 나그네)
 - 도메인을 리얼 도메인으로 쓰지 않고 강 포워딩 설정한 것 같이 보임. 정말 이게 문제인지 판단한 후 미질 지원해야 하나 고민 됨

- 장점

- 철저한 이슈 트래킹(버그,개선사항,아이디어 등)이 가능해지므로 인해 개발간 최근 문제사항을 공유하고 관리하기가 유리
- 개발소스의 변동사항 발생시 변동된 부분의 발견이 용이하고 버전 별로 추가적인 업데이트 시 개발자 상호간 정보교환이 용이
- 웹기반 으로 개발자 상호간 접근성 이 뛰어남
- 변동되는 사항은 changeset으로 명시되므로 따로 빌드 넘버 의 갱신을 하지 않아도 되며 milestone을 통해 문제점의 해결진행도가 명확히 파악 가능

- 단점

- 환경구축 시 설치해야 할 관련 프로그램이 많음
- 상용개발도구(IDE)에서는 지원되지 않음(대부분의 IDE는 자체적으로 형상관리 기능을 포함하고 있거나 툴로 존재함)
- 범용적이지 않은 db를 기본적으로 시스템의 db로 채택

- <http://dev.textcube.org/report/1>

Wiki Timeline Roadmap Browse Source **View Tickets** Search

{1} Active Tickets (285 matches)

- List all active tickets by priority.
- Color each row based on priority.
- If a ticket has been accepted, a "*" is appended after the owner's name

Max items per page 100

Update

Results (1 - 100 of 285)

1 2 3 →

Ticket	Summary	Component	Version	Milestone	Type	Owner	Status	Created
#1558	openid 공격 패치	Framework	1.9	1.9.0	defect	inureyes	assigned	02/20/2011
#1161	RSS Reader의 게시물 보기 기능 강화 요청 #2	User Interfaces	1.7	1.9.0	enhancement	inureyes	assigned	11/28/2008
#1546	1.7.9.8888 백업, 1.7.9.8888 복원시 댓글이 유실되는 문제	Core	1.7	1.7.9	defect	inureyes	assigned	01/08/2011
#1227	Windows Live Writer에서의 이상 동작	API	1.9	1.9.0	defect	daybreaker	new	03/17/2009
#1319	매킨토시 양쪽라지에서 모블로깅시 한글 문자가 해체 됨.	Plugins		1.9.0	defect	coolengineer	new	08/17/2009
#1434	미디어삽입 포스트 수정시 포스트 내용없음	User Interfaces	1.8	1.9.0	defect	graphittie	new	02/07/2010
#1289	페이지 방법 변경	User Interfaces	1.9	1.9.0	enhancement	inureyes	assigned	06/26/2009
#1542	8859 에서 데이터 복원 문제	Core	2.0	2.0.0	defect	inureyes	new	01/01/2011
#1548	2.0.0.8888 두번째 페이지로 넘어가지 않는 문제	Core	2.0	2.0.0	defect	inureyes	assigned	01/08/2011
#1430	1.8.x 로 업그레이드 시 게시물 일부가 수정이 불가능해지는 현상	Framework	1.8		defect	daybreaker	new	02/02/2010
#1590	모바일 페이지에서 다이렉트 링크 적용이 안되는 문제	User Interfaces	1.9		defect	graphittie	new	07/21/2011
#514	동기화용 티켓	Core			task	inureyes	assigned	08/12/2007
#906	스킨 index.xml 스펙 정리	Skin			task	inureyes	new	04/01/2008



Doxygen

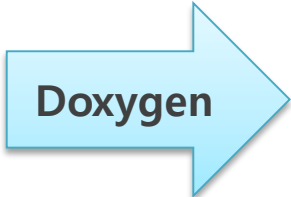
- C , C++ , JAVA 프로그램의 **주석**을 서류화한 것으로 HTML이나 워드 문서로 바꾸어 준다.

```

/**
 * @file    hello.c
 * @brief    hello world 소스파일.
 */
#include <stdio.h>

/**
 * @brief    hello Main 함수.
 * @return    성공여부.
 */
int main(
    int    argc,          /**< 인자개수 */
    char * argv[]         /**< 인자 */
)
{
    printf("Hello World\n");
    return 0;
}

```



Doxygen

hello.c 파일 참조

hello world 소스파일. [세부 사항 보기](#)

#include <stdio.h>

함수

```
int main(int argc, char *argv[])
    hello Main 함수.
```

세부 사항

hello world 소스파일.

함수 문서화

```
int main(int    argc,
        char * argv)
{
}
```

hello Main 함수.

반환값:

성공여부.

매개변수:

argc 인자개수

argv 인자

- 준비물
 - Doxygen : <http://www.doxygen.org/> 에서 다운로드
 - 소스코드
- Doxygen을 다운로드 후 별다른 설정없이 설치하면 된다.
- 최신버전(doxxygen-1.5.5)은 한글이 깨지는 문제가 있어서 구버전(doxxygen-1.4.2)을 받길 바란다.
(참고 : <http://wiki.kldp.org/wiki.php/Doxygen/>)

```

/*
 * @file    hello.c
 * @brief    hello world 소스파일.
 */
#include <stdio.h>

/*
 * @brief    hello Main 함수.
 * @return    성공여부.
 */
int main(
    int argc,          /**< 인자개수 */
    char * argv[]      /**< 인자 */
)
{
    printf("Hello World\n");
    return 0;
}

```

/* */ : 일반주석
 /** */ : Doxygen주석
 @file 파일명
 @brief 간단한 설명

함수를 문서화 하기 위한 문법이며 함수 바로 위에
 Doxygen주석을 쓴다.

@brief 간단한 설명
 @return 반환 값을 설명
 /**< */ : 인자를 설명
 ///< : 인자를 설명

hello.c 파일 참조

hello world 소스파일. [세부 사항 보기](#)

#include <stdio.h>

함수

```
int main(int argc, char *argv[])
    hello Main 함수.
```

세부 사항

hello world 소스파일.

함수 문서화

```
int main( int argc,
          char * argv[]
        )
```

hello Main 함수.

반환값:
성공여부.

매개변수:
argc 인자개수
argv 인자

```

/*
 * @file      hello.c
 * @brief     hello world 소스파일.
 */

/*
 * @mainpage  Hello World 메인페이지
 * @section  intro  소개
 * - 소개      :   프로그램의 기본을 배울수있는 프로그램.
 * @section    Program  프로그램명
 * - 프로그램명 :   Hello World 프로그램.
 * - 프로그램내용 :   화면에 Hello World!을 출력한다.
 * @section  INOUTPUT  입출력자료
 * - INPUT      :   없음.
 * - OUTPUT     :   Hello World 화면출력.
 * @section  CREATEINFO  작성정보
 * - 작성자      :   infiniterun
 * - 작성일      :   2005/04/18
 * @section  MODIFYINFO  수정정보
 * - 수정자/수정일 : 수정내역
 * - infiniterun/2005.0418 : "Helo World"에 "!"추가
 */

```

@mainpage : Main 화면에 출력될 메시지
 @section Section구분자 화면에 출력될 Section명
 — : 번호매기기 등과 같다. 한줄 한줄 설명을 쓰기 위해 쓴다.

Hello World 메인페이지

0.1

소개

- 소개 : 프로그램의 기본을 배울수있는 프로그램.

프로그램명

- 프로그램명 : Hello World 프로그램.
- 프로그램내용 : 화면에 Hello World!을 출력한다.

입출력자료

- INPUT : 없음.
- OUTPUT : Hello World 화면출력.

작성정보

- 작성자 : infiniterun
- 작성일 : 2005/04/18

수정정보

- 수정자/수정일 : 수정내역
- infiniterun/2005.0418 : "Helo World"에 "!"추가

긴 설명

```

/**
 * @brief      hello Main 함수.
 *
 * 긴 설명은 한줄을 넘긴다음 넣어준다. \n
 * 하나들.. 셋.. 넷..
 * 다섯.. 여섯.. \n
 *
 * @return     성공여부.
 */
int main(
    int argc,          /**< 인자개수 */
    char * argv[]      /**< 인자 */
)
{

```

```

int main( int  argc,
          char * argv[]
)

```

hello Main 함수.

긴 설명은 한줄을 넘긴다음 넣어준다.

하나들.. 셋.. 넷.. 다섯.. 여섯..

열거형

```

/** @brief TRUE FALSE정의. */
enum BOOLEAN
{
    FALSE=0,          /**< FALSE */
    TRUE            /**< TRUE */
};

```

```
enum BOOLEAN
```

TRUE FALSE정의.

열거형 값:

FALSE FALSE

TRUE TRUE

```

/**
 * @brief buffer structor
 *
 * Telnet에서 전송되는 데이터에 대해 프로토콜을 처리해야 하기 위하여,
 * 효율적으로 데이터를 전송해야 할 입출력 버퍼 structor
 */
struct buffer
{
    char    *buf;    /**< 데이터를 저장할 주소공간 */
    int     size;    /**< buf에 할당된 메모리 크기 */
    int     head;    /**< buf에 저장된 데이터의 처음 Index */
    int     tail;    /**< buf에 저장된 데이터의 마지막 index */
    int     count;   /**< buf에 저장된 데이터의 byte 수 */
};

```

buffer 구조체 참조

buffer structor [세부 사항 보기](#)

Data 필드

char *	buf
int	size
int	head
int	tail
int	count

세부 사항

buffer structor

Telnet에서 전송되는 데이터에 대해 프로토콜을 처리해야 하기 위하여,
효율적으로 데이터를 전송해야 할 입출력 버퍼 structor

멤버 변수 문서화

char* buffer::buf

데이터를 저장할 주소공간

int buffer::count

buf에 저장된 데이터의 byte 수

int buffer::head

buf에 저장된 데이터의 처음 Index

int buffer::size

buf에 할당된 메모리 크기

int buffer::tail

buf에 저장된 데이터의 마지막 index

```
#define MAX_READ_BUF 1024  /**< 최대 read buffer size */  
short port;                /**< Telnet port number */
```

#define 문서화

```
#define MAX_READ_BUF 1024
```

최대 read buffer size

변수 문서화

```
short port
```

Telnet port number

```

/**
 * @file      hello.c
 * @brief      hello world 소스파일.

파일여러줄 설명입니다.\n
전자 여러줄 입니다.\n
음.. 하나. 둘. 셋
넷다섯.
 */

/**
 * @mainpage      Hello World 메인페이지
 * @section intro 소개
 * - 소개      :   프로그램의 기본을 배울수있는 프로그램.
 * @section      Program 프로그램명
 * - 프로그램명 :   Hello World 프로그램.
 * - 프로그램내용 :   화면에 Hello World!을 출력한다.
 * @section INOUTPUT   입출력자료
 * - INPUT      :   없음.
 * - OUTPUT     :   Hello World 화면출력.
 * @section CREATEINFO   작성정보
 * - 작성자     :   infiniterun
 * - 작성일     :   2005/04/18
 * @section MODIFYINFO   수정정보
 * - 수정자/수정일 :   수정내역
 * - infiniterun/2005.0418 :   "Hello World"에 "!"추가
 */

#include <stdio.h>

#define MAX_READ_BUF 1024 /**< 최대 read buffer size */

short port; /**< Telnet port number */

```

```

/**
 * @brief buffer structor

Telnet에서 전송되는 데이터에 대해 프로토콜을 처리해야 하기 위하여,
효율적으로 데이터를 전송해야 할 입출력 버퍼 structor
 */
struct buffer
{
    char *buf; /**< 데이터를 저장할 주소공간 */
    int size; /**< buf에 할당된 메모리 크기 */
    int head; /**< buf에 저장된 데이터의 처음 Index */
    int tail; /**< buf에 저장된 데이터의 마지막 index */
    int count; /**< buf에 저장된 데이터의 byte 수 */
};

/** @brief TRUE FALSE정의. */
enum BOOLEAN
{
    FALSE=0, /**< FALSE */
    TRUE /**< TRUE */
};

/**
 * @brief      hello Main 함수.

긴 설명은 한줄을 넘긴다음 넣어준다. \n
하나둘. 셋.. 넷..
다섯.. 여섯.. \n

@return      성공여부.
 */
int main(
    int argc, /**< 인자개수 */
    char *argv[] /**< 인자 */
)
{
    printf("Hello World!\n");
    return 0;
}

```

03 주석 사용 포맷 by RTLAB

Function Documentation

```
int pushStack ( STACK * stack,  
               void * dataInPtr  
               )
```

Push in Stack

Parameters:

stack stack pointer
dataInPtr data pointer

Remarks:

contents

Returns:

ture(successful) or false(underflow)

See also:

reference

Example

```
// Explane  
#include <stdio.h>
```

Output

..?

```
/**  
 * Push in Stack  
 * @param stack stack pointer  
 * @param dataInPtr data pointer  
 * @remark  
 *     contents  
 * @return ture(successful) or false(underflow)  
 * @sa  
 *     reference  
 * @par Example  
 * @code  
 *     // Explane  
 *     #include <stdio.h>  
 * @endcode  
 * @par Output  
 *     ..?  
 */  
int pushStack(STACK * stack,void * dataInPtr)  
{  
    STACK_NODE * newPtr;  
  
    newPtr = (STACK_NODE *) malloc(sizeof(STACK_NODE));  
    if(!newPtr)  
        return 0;  
  
    newPtr->dataPtr = dataInPtr;  
  
    newPtr->link = stack->top;  
    stack->top = newPtr;  
  
    (stack->count)++;  
  
    return 1;  
}
```

- Open Source 개발환경과 글로벌 표준 개발 기법(CICD)으로 무장하여 IoT 소프트웨어를 개발하자!
- 많은 Open Source 기반 개발환경(도구) 사용법을 일상화하여 글로벌 표준에 준하는 소프트웨어 품질을 보장할수 있는 소프트웨어(펌웨어)엔지니어가 되자!
- Open Source로 공개된 IoT 요소 기술(예, 통신 스택, 미들웨어등)은 매우 위험하다.
- 자기가 개발하지 않은 것을 개발한 것 처럼 위장(포장) 하지 말자.



Thank you