

시스템 이슈 식별과 해결을 위한 진단 Framework 구축 사례 연구

신현중*, 박동현**

* LG CNS S/W공학센터 테스트팀

A case study on the diagnostic framework in order to verify and solve a system issue

Shin Hyun-Jong*, Park Dong-Hyun

Testing Team, S/W Engineering Center, LG CNS

E-mail : hjongshin@lgcns.com, dnghpark@lgcns.com

요 약

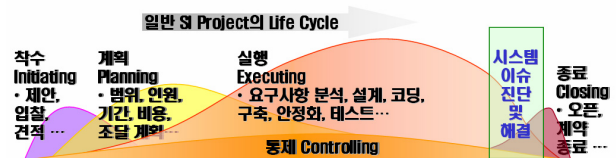
본 연구는 SI 프로젝트 현장에서 시스템의 주요 이슈를 최소화하기 위한 진단 Framework를 구축하는 방안에 대해 고찰한 것이다. SI 프로젝트에서 심각한 Risk로 발전할 가능성이 있는 시스템 이슈들을 효과적으로 관리하기 위해서는, 사전에 ① 시스템 이슈 점검을 위한 조직에서 이슈 식별을 위한 방법과 절차를 확립해야 하고, ② 프로젝트 수행 조직의 구성원이 담당 업무 별로 이슈 식별을 할 수 있도록 역할을 설정해야 한다. 더불어 ③ 식별된 이슈들은 올바른 절차에 따라 시정 조치가 취해져야 하고, ④ 시정 조치가 합당한지에 대한 영향 분석 및 평가를 거쳐 시스템 오픈 여부를 결정할 수 있어야 한다. 뿐만 아니라 ⑤ 유사한 SI 프로젝트에서 발생 가능한 시스템 이슈들은 발생 원인과 해결 방법이 정리되어 시행착오를 줄일 수 있도록 진단 방법론이 지속적으로 보강되어야 한다.

1. 서론

성공적으로 SI 프로젝트를 완수하기 위해서는 무엇보다도 최종 단계의 Output인 시스템이, 심각한 이슈가 발행하지 않고 안정적으로 운영되어야 한다. 프로젝트 관리 영역인 ① 통합관리, ② 범위관리, ③ 시간관리, ④ 원기관리, ⑤ 품질관리, ⑥ 인적자원관리, ⑦ 의사소통관리, ⑧ 위험관리, ⑨ 조달관리 등 전체 영역에서 합리적인 관리 방안이 마련되어 있어야 함은 물론 ⑧ 위험 관리 차원에서 구축한 시스템 아키텍처에 적합한 이슈 진단 및 해결 방안을 마련할 필요성이 있다.

프로젝트 생명 주기(Project Life Cycle) 방법론에 따라 착수, 계획, 실행, 종료 단계를 수행하던, SDLC(Software Development Life Cycle) 방법론에 따라 타당성 조사, 요구 분석, 설계, 개발, 시험, 유지보수, 폐기 단계를 이행하던 간에, 각 단계 별로 점검(Check)이나 통제(Control)를 통해 단계 별 이슈를 식별하여 해결 또는 개선하는 작업은 프로젝트 각 단계에서 효율적으로 위험을 제거하고 작업의 완성도를 높여주는 역할을

수행한다. 그러나 말 그대로 시스템 통합(SI=System Integration) 프로젝트에서 아무리 분산된 개별 단계에서 적시에 점검이나 통제를 효과적으로 수행했다고 해도, 완전하게 통합된 시스템에 대해 다수의 최종 사용자 관점에서 이슈를 점검하지 않는다면, 완성된 시스템의 연속성과 안정성을 보증할 수 없는 문제가 발생하게 된다.



[그림01] 시스템 이슈 진단 및 해결 Task

이런 이유로 반복 혹은 공통으로 발생하는 주요 이슈들은 각기 SI 프로젝트에서 Checklist처럼 항목 별로 세분화해서

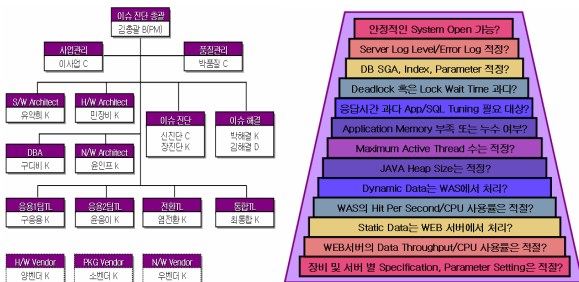
점검되어야 하며, 효과적으로 대처 방안을 수립하기 위해서는 Domain 별로 구분하여 기술적인 관점에서 통제할 필요성이 있다.

따라서 SI 프로젝트에서는 시스템 통합이 거의 완료된 단계에서 시스템을 진단하고 위험으로 발전할 가능성이 있는 부분에 대해서는 해결 조치를 취해야 하며, 본 논문에서는 이를 “시스템 이슈 식별과 해결을 위한 진단 Framework”로 지칭하고 이에 대해 고찰해 보고자 했다.

2. 본론

2-1. 이슈 진단 TFT 조직 구성

시스템 통합이 거의 완료된 SI 프로젝트에서 효과적으로 시스템 이슈 식별과 해결을 위한 진단 Framework를 구축하기 위해서는, 프로젝트 외부의 조직과 내부의 조직이 연계된 TFT(Task Force Team)을 구성할 필요성이 있다.

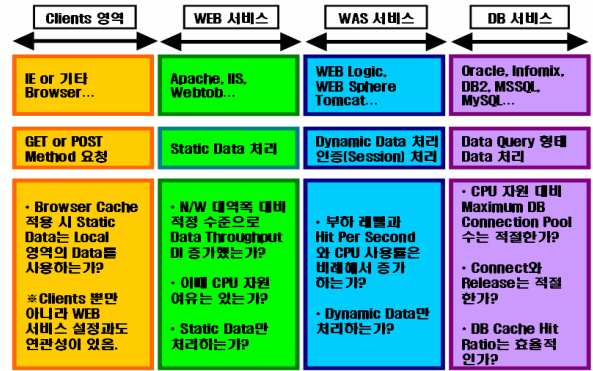


[그림02] 이슈 진단 조직과 Domain별 주요 이슈

먼저 프로젝트 외부 조직은 TFT를 Leading할 수 있는 프로세스의 틀(Frame)을 가지고 있어야 하며 대상 시스템을 분석하여 발생 가능한 이슈 목록과 이를 해결할 담당자를 프로젝트 내부 조직의 인원을 할당해야 한다. 예를 들어 시스템 이슈 진단을 위한 외부 조직은 주요 시스템의 O/S Parameter 설정 이상 - H/W Architect, 네트워크 장비의 부하 처리 이상 - N/W Architect, System Layer별 역할 분리 수행 여부 및 AP Parameter Setting - Application Architect, 주요 업무별 Application 또는 SQL 이상 - 대상 업무 개발자, DB Parameter 설정 및 Deadlock 발생 여부 - DBA 등 발생 가능한 이슈 목록과 프로젝트 조직 내 담당자를 참여하도록 연결해 놓아야 하며, 자체적인 해결이 어려울 경우 외부의 이슈 해결 조직이 SWAT 형태로 투입될 수 있도록 준비할 필요성이 있다.

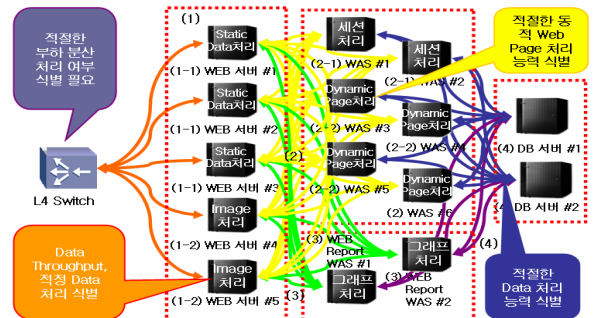
2-2. 시스템 아키텍처 계층별 이슈 진단

최근의 시스템 환경은 과거 Package 적용 3 Tiers C/S 환경과 크게 상이한 WEB 기반 환경으로 전환된 상태이다. 과거의 Package 환경의 경우 Clients Tier 즉 전용 단말기 혹은 장비에 대다수의 Application Component를 탑재하여, Application Layer의 경우 Clients의 Application 배포 정도로 그 역할이 축소되어 있었다.



[그림03] WEB 기반 시스템의 아키텍처 계층별 진단

그러나 WEB 기반 시스템은 Browser 등으로 Clients Program이 표준화된 상태이며, 사용자가 필요한 작업을 수행할 때마다 주요한 Request를 Application Layer로 전송하는 형태로 변화했으며, 특히 Application Layer는 ① Static Data를 처리하는 WEB 서비스 Layer, ② Dynamic Data를 처리하는 WAS(WEB Application Server) Layer로 분화되었다. 더불어 각 System Architecture Layer는 폭포수 모델(Waterfall)과 같이 앞의 Layer에서의 처리가 원활하게 이루어져야만 뒤 Layer에 처리가 전달되므로 이런 특성을 고려하여 진단 케이스 혹은 시나리오를 설계해야 한다.



[그림04] 계층별 역할을 고려한 진단 케이스 설계

System Architecture Layer 관점에서의 점검 Task는 각 Layer별 Device의 적절한 구성, O/S 설정에서부터 Service Program의 Parameter 설정까지 Bottom Up 방식으로 시스템의 적절성을 확인해야 한다.

2-3. 응용 및 모듈/업무 별 이슈 진단

시스템 아키텍처 관점의 이슈 진단이 완료된 이후에는 주요한 모듈과 업무에 대한 이슈 진단이 수행되어야 한다. 주요한 모듈은 사용자 세션 처리 및 인증 관리 모듈, 권한 별 메뉴 화면 관리 모듈 등이며 주요 업무는 URL 입력 후 초기화면, 로그인/로그아웃, 메인 메뉴 및 화면 등 업무 별 발생빈도, 사용자수, 특정시점 집중사용, 복잡도, 고객 요청 등을 반영하여 다수의 사용자가 대량 작업을 병렬로 수행할 때 안정적인 기능 수행, 에러 발생, 메모리 누수, 세션 미 반환 여부 등을 내구성 시험(Durability Test) 관점에서 점검해야 한다.

더불어 부품 공정 처리, MES, 물류 시스템과 같이 대상

시스템의 데이터가 흐름이 있는 경우, 흐름에 따라 대량 작업을 재현한 상태에서 CRUD(Create, Retrieve, Update, Delete) 작업 수행 시 올바르게 Primary Key 속성 Data이 발급되는가, Database에서 Deadlock이나 과도한 Lock Wait Time이 발생하는가, Data 정합성 문제가 발생하는가 등에 대해 점검을 수행해야 한다.

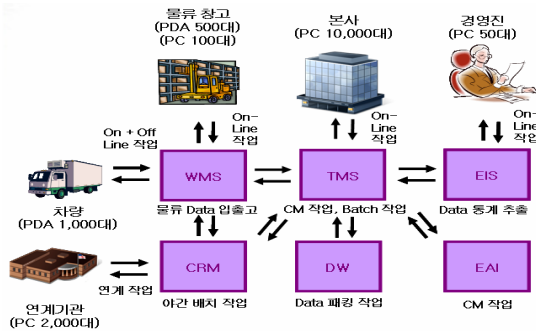


[그림05] Data 흐름이 있는 시스템의 진단

2-4. 통합 시스템 관점의 이슈 진단

장비 구입과 운영 환경 설정, O/S를 비롯한 주요 S/W 설치 및 Patch, 모듈 별로 개발이 완료된 Application 적재, 전환이 완료된 Data 적재 등 통합 시스템의 구축이 완료된 후 통합 테스트를 통해 시스템의 주요 결함을 제거한 후 통합 시스템을 Open하기 이전에 반드시 리허설을 거칠 필요성이 있으며, 이때가 통합 시스템에 대한 이슈를 진단할 수 있는 최적의 기회가 된다.

반드시 운영 환경 및 이에 준하는 환경에서 Peak Time의 On Line, Off Line, CM(Concurrent Managing), 배치, EAI 등 전 영역의 적정 부하를 시스템에 생성한 뒤 시스템 이슈를 진단할 필요성이 있다.



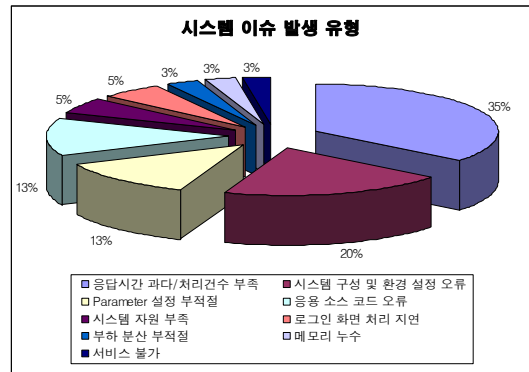
[그림06] 연계를 포함한 통합 시스템 관점의 진단

이때의 이슈 진단 기본 점검 항목은 ① 목표 처리량(TPS) 달성도, ② 90% 응답 시간 만족 비율, ③ 적정 시스템 사용률 충족 여부이며, 이와 별도로 주요 측정 요소 별 장애 및 오류 발생 여부를 측정할 필요성이 있다. Peak Time의 적정 부하 레벨에서 시스템이 이상 없이 동작할 경우 적정 부하의 배수 부하를 생성하여, 과부하시의 시스템 상황 또는 시스템의 한계 처리 상황을 측정하여, 병목 발생 구간을 식별한 뒤 사용자 증가 등의 상황에 대한 대비책을 마련할 수 있다.

3-1. 사례 별 이슈 및 해결 내역

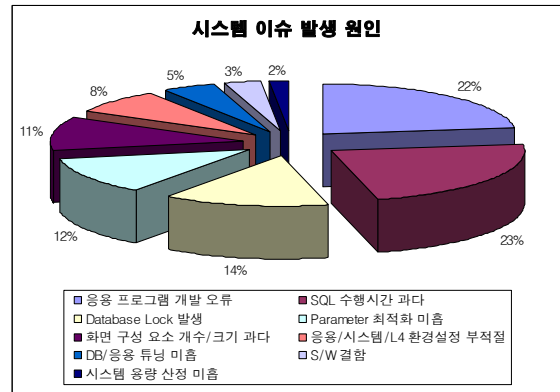
2008년 LG CNS에서 구축한 프로젝트 중 시스템 진단을

수행한 뒤 필수 시정 사항이 발견된 44건의 프로젝트의 시스템 이슈 유형을 분석한 결과, 가장 빈도수가 높은 항목을 순서대로 열거하면 응답시간 과다/처리건수 부족 - 39건(35%), 시스템 구성 및 환경 설정 오류 - 22건(20%), Parameter 설정 부적절 - 14건(13%), 응용 소스 코드 오류 - 14건(13%), 시스템 자원 부족 - 6건(5%), 로그인 화면 처리 지연 - 6건(5%), 부하 분산 부적절 - 3건(3%), 메모리 누수 - 3건(3%), 서비스 불가 - 3건(3%)인 것으로 나타났다.



[그림07] 이슈 발생 유형 분석

시스템에서 발생한 이슈의 주요 원인은 다음과 같이 응용 프로그램 개발 오류 - 15건(23%), SQL 수행 시간 과다 - 15건(23%), Database Lock 발생 - 9건(14%), Parameter 최적화 미흡 - 8건(12%), 화면 구성 요소 개수/크기 과다 - 7건(11%), 응용/시스템/L4 환경설정 부적절 - 5건(8%), DB/응용 튜닝 미흡 - 3건(5%), S/W 결함 - 2건(3%), 시스템 용량 산정 미흡 - 1건(2%) 등으로 나타났다.



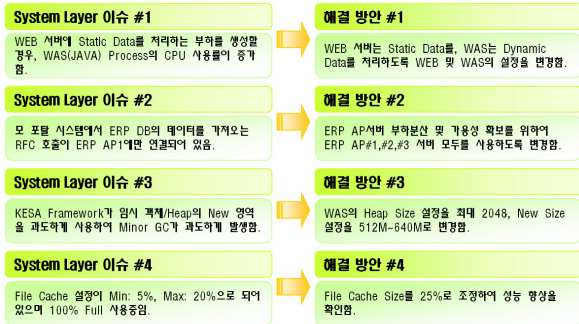
[그림08] 이슈 발생 원인 분석

3-2. 이슈 별 해결 방안

이슈 해결 방안은 ① System ② Application ③ Database의 3가지 Layer 별로 구분하였다.

먼저 System Layer의 이슈 해결 내역으로는 WEB 서버에 대량의 Static Data를 처리하도록 했을 때 WAS 서비스 Program의 CPU 사용률이 증가하여, WAS는 Dynamic Data만을 처리하도록 하고 WEB은 Static Data를 처리하도록 WAS 프로그램의 Parameter 설정을 변경했다.

또한 File Cache의 사용량이 현재 설정 값의 최대치에 도달하여 Kernel Parameter를 더 크게 증가시켜 이슈를 해결했다.



[그림09] System Layer 이슈 해결 내역

Application Layer의 이슈 해결 내역으로는, DB Connection을 Release하지 않고 Execute Query()를 두 번 호출하여 Connection Pool 을 과도하게 사용하는 현상이 발생하였으나 Execute Query()를 한번만 호출하도록 소스를 수정하여 해당 문제를 해결했으며, 일부 소스 코드에서 Exception에 대한 처리가 미흡하여 오류가 과도하게 누적되는 Application을 찾아 일괄 수정하여 문제를 해결했다.



[그림10] Application Layer 이슈 해결 내역

Database Layer의 이슈 해결 내역으로는 DB Buffer Cache의 Hit Ratio가 낮고 Physical Disk 의 I/O가 많이 발생함에 따라 DB Buffer Cache 크기를 증가시킴으로써 Hard Disk에서 읽은 데이터의 재 사용률을 높여 성능을 향상시켰으며 응답 시간이 과도하게 느린 업무의 SQL을 분석하여 Sort Merge 조인으로 실행 계획이 수립된 것을 Nested Loop 조인으로 변경하여 온라인 업무의 응답 시간 단축시켰다.



[그림11] Database Layer 이슈 해결 내역

4. 결론

위험으로 발전할 가능성이 있는 이슈 들을 사전에 예방하기 위해서는, 잘 만들어진 프로세스와 체계적인 품질관리 방안을 적용하여 관리하는 것도 중요하지만 시스템 통합이 완료된 이후 점검 활동을 통해 이슈들을 사전에 식별하고, 해결할 수 있는 효과적인 시스템 진단 방법에 대한 Framework를 구성하는 것도 반드시 필요하다.

특히 시스템 통합이 완료된 이후의 한정된 기간 안에 체계적으로 시스템 진단 업무를 수행하기 위해서는 유사한 시스템에서 자주 발생하는 이슈들과 그 해결책을 축적시킨 시스템 진단 Framework를 만들고 지식 관리 및 공유 차원에서 지속적인 사례 연구를 통해 방법론을 보강할 필요성이 있다고 판단된다.

참고문헌

- [1] IT S/W 기술분야의 표준화 동향분석 및 표준화 연구, IT S/W 기술위원회, 한국정보통신기술협회, 2003
- [2] PMBOOK 3rd Edition, PMI, PMI 개정됨, 2004
- [3] 정보시스템 감리방법론 연구, 정보화평가분석단 감리연구부, 한국전산원, 2000
- [4] 소프트웨어 품질 인증 사업에 관한 연구, 정보통신기술협회, 정보통신부, 2001
- [5] S/W 사용성 개선을 위한 평가 모델 및 방법론에 관한 연구, 상명대학교, 한국소프트웨어진흥원, 2005
- [6] S/W 품질 향상을 위한 표준 연구, 전자통신연구원, 정보통신부, 2000
- [7] SI 환경하에서 품질관리가 프로젝트 성공에 미치는 영향에 관한 연구, 문수현, 건국대 정보통신대학원 석사 학위 논문, 2006
- [8] PM이 꼭 알아야 할 기술 Risk 관리, 리스크진단팀, LG CNS, 2004
- [9] 워크로드 중심의 성능 관리와 용량 계획에 관한 연구, 이원용, 숭실대 정보과학대학원 정보통신학 석사학위 논문, 2003 외 다수...