



프로젝트 관리론

- 기능점수에 의한 비용예측 (1) -

2005. 10

홍 장 의

jehong@chungbuk.ac.kr

목 차

- 소프트웨어 생산성과 척도
- 소프트웨어 비용산정 기법
- 정통부 고시 : 제2004-8호
- 정리 및 토의

소프트웨어 생산성과 척도

Software Engineering - Project Management

Needs

- Project failure due to ineffective management
- Project-late, unreliable, over budget, poor performance

Differences from other types of engineering

- Product is intangible
- Not have a clear understanding of the S/W process
- Large S/W systems are often 'one-off' projects

Most Important Contributor to a Successful Software Project

" .. it's not the tools that we use, but it's the people"

"... having smart people .. very little else matters in my opinion."

" The only rule I have in management is to ensure I have good people"

Management Steps

Step 1. Planning

- Understanding & documenting the goal.
- Developing a schedule, budget and other resource req.s.

Step 2. Acquisition of resources

- Space, computing resources, materials and human resources

Step 3. Execution

- Putting the plan into action.

Step 4. Monitoring:

- Checking the progress of the project.
- Taking necessary actions to handle deviation from the plan.

Software Productivity

For project planning, need to

- Estimate the difficulty of the task.
- Estimate how much of the task each engineer can solve.

Productivity metrics

- Amount of functionality
- LOC: not an ideal productivity metrics

How to quantify the concept of functionality?

- Function points.

Function Points

Attempt to quantify the functionality of a software system.

Characterize the complexity of the system.

Can be used to forecast

- how long it will take
- how many people will be needed to do it.

Suitable for information processing applications.

Can measure productivity, amount of money, number of errors.

Can be used as bases for future planning.

Used to measure the relative power of different languages.

Many problems to be solved, not promising.

Code Size (LOC)

The most commonly used metrics to measure productivity.

Two most common code size:

- DSI(delivered source instructions)
 - Only lines of code delivered to the customer.
- NCSS(non-commented source statements)
 - Comment lines are not counted.

Many problems to be solved, but easy to measure.

Factors Affecting Productivity

The capability of the personnel

The complexity of the product

Required reliability

- Timing constraints(in real-time systems)

Schedule constraints

Language experience

Personnel turnover

Cancelled projects

Reorganizations

Restructuring of the systems

소프트웨어 비용산정 기법

Techniques of S/W Cost Estimation (Boehm)

Algorithmic cost modeling : COCOMO

Expert judgment

Estimation by analogy

Parkinson's Law : determined by available resources

Pricing to win

Top-down estimation : overall functionality, first

Bottom-up estimation

Cost Estimation

Primary cost in software engineering is for people.

Two uses in software project management:

- Decide how many engineers needed
- Assess the progress of the project

Need metrics

- Structural metrics: needs a program.
- Predictive metrics: LOC for total life cycle costs.

COCOMO Model (Constructive Cost Model) : Intermediate

Divide projects into three categories:

- Organic mode
- Semi-detached mode
- Embedded mode

$$PM = C * KLOC \exp(k)$$

PM: person month

TDEV: months required to complete the project

| Mode | Effort | Schedule |
|---------------|----------------------------|----------------------------|
| Organic | $PM=3.2*(KDSI) \exp(1.05)$ | $TDEV=2.5*(PM) \exp(0.38)$ |
| Semi-detached | $PM=3.0*(KDSI) \exp(1.12)$ | $TDEV=2.5*(PM) \exp(0.35)$ |
| Embedded | $PM=2.8*(KDSI) \exp(1.20)$ | $TDEV=2.5*(PM) \exp(0.32)$ |

COCOMO Software Development Model[Boehm]

| Feature | Mode | | |
|--|-----------|---------------|--------------|
| | Organic | Semi-detached | Embedded |
| Organizational understanding of product objectives | Through | Considerable | General |
| Experience in working with related software system | Extensive | Considerable | Moderate |
| Need for software conformance with pre-established requirements | Basic | Considerable | Full |
| Need for software conformance with external interface specifications | Basic | Considerable | Full |
| Concurrent development of h/w and operational procedures | Some | Moderate | Extensive |
| Need for innovative data processing architectures, algorithms | Minimal | Some | Considerable |
| Premium on early completion | Low | Medium | High |
| Product size range | <50KDSI | <300KDSI | All size |

COCOMO Software Development Model[Boehm]

| | Mode | | |
|----------|-----------------------|--|--|
| | Organic | Semi-detached | Embedded |
| Examples | Batch data reduction | Most transaction process systems | Large, complex, transaction processing systems |
| | Scientific models | New OS DBMS | Ambitious, Very large OS |
| | Business models | Ambitious, inventory, product control | Avionics |
| | Familiar OS, compiler | Simple command control | Ambitious Command control |

Effort Multipliers in COCOMO Intermediate Model

| Cost Driver | Rating | | | | | |
|------------------------------------|----------|------|---------|------|-----------|------------|
| | Very low | Low | Nominal | High | Very high | Extra high |
| Product attributes | | | | | | |
| Required software reliability | .75 | .88 | 1.00 | 1.15 | 1.40 | |
| Data base size | | .94 | 1.00 | 1.08 | 1.16 | |
| Product complexity | .70 | .85 | 1.00 | 1.15 | 1.30 | 1.65 |
| Computer attributes | | | | | | |
| Execution time constraints | | | 1.00 | 1.11 | 1.30 | 1.66 |
| Main storage constraints | | | 1.00 | 1.06 | 1.21 | 1.56 |
| Virtual machine volatility | | .87 | 1.00 | 1.15 | 1.30 | |
| Computer turnaround time | | .87 | 1.00 | 1.07 | 1.15 | |
| Personnel attributes | | | | | | |
| Analyst capability | 1.46 | 1.19 | 1.00 | .86 | .71 | |
| Application experience | 1.29 | 1.13 | 1.00 | .91 | .82 | |
| Programmer capability | 1.42 | 1.17 | 1.00 | .86 | .70 | |
| Virtual machine experience | 1.21 | 1.10 | 1.00 | .90 | | |
| Programming language experience | 1.14 | 1.07 | 1.00 | .95 | | |
| Project attributes | | | | | | |
| Use of modern programming practice | 1.24 | 1.10 | 1.00 | .91 | .82 | |
| Use of software tools | 1.24 | 1.10 | 1.00 | .91 | .83 | |
| Required development schedule | 1.23 | 1.08 | 1.00 | 1.04 | 1.10 | |

Pragmatics of the Cost Drivers

Required s/w reliability

- very low: when the defects just need to be removed by the developers without any other consequences.
- nominal: when the defects indicate a moderate loss.
- very high: when there is a possible loss of human life.

Data base size

- Ratio of the size of the database to the program size
- low: 1
- nominal: 10
- high: 100
- very high: 1000

Product complexity

- very low: if the modules are made of simple mathematical expressions.
- extra high: if the modules are dealing with multiple resource scheduling.

Pragmatics of the Cost Drivers (Cont.)

Execution time constraint

- The percentage of available execution time
- (when the program has an execution time constraints)
- nominal: 50 %
- extra high: 95%

Main storage constraint

- Volume of main storage used by the program.
- nominal: use 50% of the main storage
- extra high: use 95% of the main storage

Pragmatics of the Cost Drivers (Cont.)

Virtual machine volatility

- The frequency of changes on h/w & s/w on which the program is developed.

Computer turnaround time

- The computer response time as seen by the programmer.
- The longer response time, the higher is the manpower cost.
- low: interactive systems
- very high: more than 12 hours

Pragmatics of the Cost Drivers (Cont.)

Analyst capability

- Team analysis ability, efficiency and ability to co-op.

Application experience

- very low: less than 4 months experience
- very high: greater than 12 years experience

Programmer capability

- Similar to analyst capability,
- but applicable to programmers as a team.

Virtual machine experience

- Familiarity with the host processor
- very low: less than one month experience

An Example: The Module Controller Project

Develop the software used to control the switching of the subscribers in a digital switching subscribers system of a local telephone exchange.

An expected size of the code was estimated as 30K, but from 25.5K to 34.5K variance observed.

Let us start cost estimation of this project.

First, select the program mode:

Second, select cost drivers:

- Not possible to select a unique rating for most of the cost drivers, thus decide to a pair if necessary.

The Module Controller Project (Cont.)

Reliability:

- requires a long time between failures: high 1.15
- for security reasons: very high 1.40

Data base size:

- initially uncertain,
- likely to be nominal, 1.00
- could be very high, 1.16

Product complexity:

- extra high, 1.65

Execution time constraint:

- no timing constraints was apparent, nominal 1.00

The Module Controller Project (Cont.)

Main storage constraint:

- no excessive constraints,
- nominal 1.00
- could be 1.06

Virtual machine volatility:

- promised that the host processor would be normally operating, low 0.87
- later found that the target processor was also used for hardware debugging, very high 1.30

Computer turnaround time:

- at the start of the project, nominal 1.00
- later became apparent that the host mainframe had a limited availability because of increased demand and failure thus, very high 1.07

The Module Controller Project (Cont.)

Analyst capability:

- The analysts are capable, but formed recently, nominal 1.00

Application experience:

- There was a kernel of experienced analysts, but more software engineers need to be recruited.
- No experience in other members of the team.
- low 1.03
- high 0.91

Programmer capability:

- Most programmers need to be recruited
- low 1.17
- nominal 1.00

The Module Controller Project (Cont.)

Virtual machine experience:

- No one was familiar with the new host machine.
- very low 1.21

Programming language experience:

- New language will be used.
- very low 1.14

Modern programming practices:

- Use of modern programming techniques just started.
- low 1.00

Use of software tools:

- Going to be supplied with adequate tools, but soon discovered that new tools still had a few internal problems.
- nominal 1.00
- low 1.00

The Module Controller Project (Cont.)

Schedule constraints

- rated nominal at first, 1.00
- but needed to spend time in training, thus had to accelerate the development, low 1.08

Third, calculate cost multipliers:

- P, pessimistic: high cost driver values
- O, optimistic: low cost driver values
- $P = 1.40 * 1.16 * 1.65 * 1.00 * 1.06 * 1.30 * 1.07 * 1.00 * 1.13 * 1.17 * 1.21 * 1.14 * 1.10 * 1.10 * 1.08 = 9.42$
- $O = 1.15 * 1.00 * 1.65 * 1.00 * 1.00 * 0.87 * 1.00 * 1.00 * 0.91 * 1.00 * 1.21 * 1.14 * 1.10 * 1.00 * 1.00 = 2.28$
- Calculate the average rating, M
- M: obtained by calculating the average for each rating.
- $M = 1.27 * 1.08 * \dots * 1.05 * 1.04 = 4.73$

The Module Controller Project (Cont.)

Fourth, calculate manpower cost:

- Use the equation $PM = 2.8 * (KDSI) \exp(1.20)$
- Size(KDSI) PM O(PM) M(PM) P(PM)

| Size(KDSI) | PM | O(PM) | M(PM) | P(PM) |
|------------|-----|-------|-------|-------|
| 25.5 | 136 | 310 | 643 | 1281 |
| 30.0 | 166 | 378 | 785 | 1564 |
| 34.5 | 196 | 447 | 927 | 1846 |

The Module Controller Project (Cont.)

Fifth, Calculate the development time (TDEV)

- Use the equation $TDEV = 2.5 * (PM) \exp(0.32)$
- ex) $2.5 * 785 \exp(0.32) = 21.1$ (months)
- $785/21.1 = 37.2$ person needed for 21.1 months.

| Size(KDSI) | PM | O(PM) | M(PM) | P(PM) |
|------------|-----|-------|-------|-------|
| 25.5 | 136 | 15.7 | 19.8 | 24.7 |
| 30.0 | 166 | 16.7 | 21.1 | 26.3 |
| 34.5 | 196 | 17.6 | 22.2 | 27.7 |

정보통신부 고시 제2004-8호
소프트웨어사업대가의 기준

정통부 고시 : 제2004-8호

소프트웨어 사업 발주시 예정가격 산정과 소프트웨어사업자가 국내에서 수주하는 소프트웨어 사업의 적정한 대가 산정 기준

소프트웨어 사업비

- 소프트웨어 개발비
- 소프트웨어 유지보수비
- 시스템 운영 환경 구축비
- 데이터베이스 구축비
- 자료 입력비
- 정보전략계획수립비

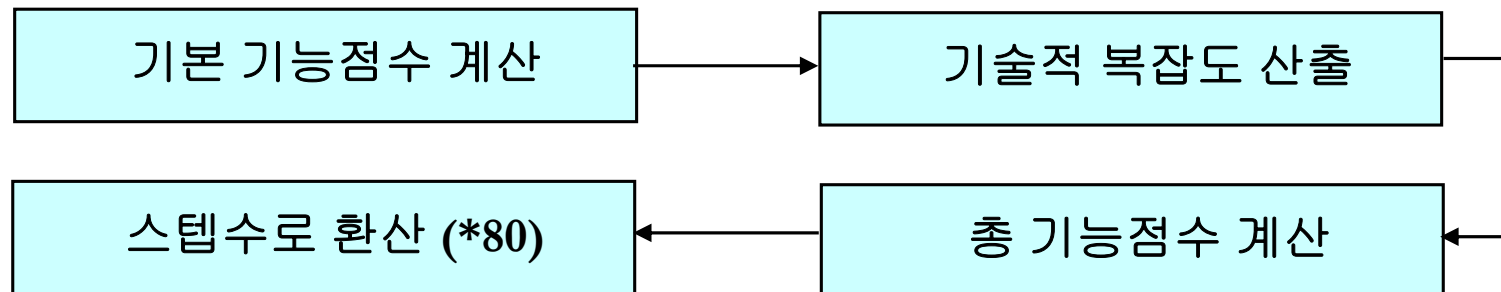
기능점수

- 소프트웨어 크기를 결정하는 소프트웨어 기능유형별 수량과 성능 및 품질요인들의 영향도를 고려하여 계산되는 수치

정통부 고시 : 제2004-8호 (Cont.)

개발비 규모 (스텝수)의 산정 방식

- 본수에 의한 스텝수 산정
 - 총 스텝수 = 개발 소프트웨어 본수 X 정보처리 형태별 평균스텝수
- 직접 산정방식
 - 통신, 제어통제, 장비내장형 시스템 등 본수 산정이 어려운 경우
- 기능 점수
 - 일반 사무용, 소규모 시스템 개발시 권장
 - 산정 절차



정통부 고시 : 제 2004-8 호 (Cont.)

소프트웨어 개발의 단계 및 공정 : 분석 단계

| 단계 | 공정 | 활 동 |
|----|------------|--|
| 분석 | 공정구현 | <ul style="list-style-type: none"> ·생명주기 모형 정의/선정 ·지원 공정 구현 ·표준, 방법, 도구, 언어 등의 선정/조정/사용 ·개발계획 개발 ·비인도 품목 식별 |
| | 시스템 요구분석 | <ul style="list-style-type: none"> ·시스템 요구명세 작성 ·시스템 요구사항 평가 |
| | 소프트웨어 요구분석 | <ul style="list-style-type: none"> ·소프트웨어 요구명세 작성 ·소프트웨어 요구사항 평가 ·합동검토 실시 및 베이스라인 설정 |

정통부 고시 : 제 2004-8 호 (Cont.)

소프트웨어 개발의 단계 및 공정 : 설계 단계

| 단계 | 공정 | 활 동 |
|----|------------|---|
| 설계 | 시스템 구조설계 | <ul style="list-style-type: none"> . 시스템 구조 설정 . 시스템 구조 및 항목 요구사항 평가 |
| | 소프트웨어 구조설계 | <ul style="list-style-type: none"> . 소프트웨어 구조 설정 . 최상위 수준 외부 인터페이스 개발 . 최상위 수준 DB 설계 . 사용자 문서 초판 개발 . 소프트웨어 통합시험 요구사항 및 일정 정의 . 소프트웨어 항목의 구조, 인터페이스, DB 설계 평가 . 합동검토 실시 |
| | 소프트웨어 상세설계 | <ul style="list-style-type: none"> . 소프트웨어 구성품 상세설계 . 인터페이스 상세설계 . DB 상세설계 . 사용자 문서 갱신 . 유닛 시험 요구사항 및 일정 정의 . 소프트웨어 통합시험 요구사항 및 일정 갱신 . 상세설계 및 시험 요구사항 평가 . 합동검토 실시 |

정통부 고시 : 제 2004-8 호 (Cont.)

소프트웨어 개발의 단계 및 공정 : 구현 및 시험 단계

| 단계 | 공정 | 활 동 |
|----|---------------|--|
| 구현 | 소프트웨어 코딩 및 시험 | <ul style="list-style-type: none"> .유니트와 DB 코딩 및 시험절차와 데이터 개발 .유니트 및 DB 시험 .사용자 문서 갱신 .소프트웨어 통합시험 요구사항 및 일정 갱신 .소프트웨어 코드와 시험결과 평가 |

| 단계 | 공정 | 활 동 |
|----|----------|---|
| 시험 | 소프트웨어 통합 | <ul style="list-style-type: none"> .통합계획 개발 .유니트와 구성품 통합 및 시험 .사용자 문서 갱신 .소프트웨어 자격시험 준비 .통합계획, 설계, 코드, 시험, 시험결과, 사용자 문서 평가 .합동검토 실시 |

정통부 고시 : 제 2004-8 호 (Cont.)

소프트웨어 개발의 단계 및 공정 : 시험 단계

| 단계 | 공정 | 활 동 |
|----|------------|--|
| 시험 | 소프트웨어 자격시험 | <ul style="list-style-type: none"> . 소프트웨어 자격시험 실시 . 사용자 문서 갱신 . 설계, 코드, 시험, 시험결과, 사용자 문서 평가 . 감사 지원 . 인도 소프트웨어 준비 및 베이스라인 설정 |
| | 시스템 통합 | <ul style="list-style-type: none"> . 통합 및 시험 . 시스템 자격시험 준비 . 통합 시스템 평가 |
| | 시스템 자격시험 | <ul style="list-style-type: none"> . 시스템 자격시험 실시 . 시스템 평가 . 감사 지원 . 인도 소프트웨어 준비 및 베이스 라인 설정 |
| | 소프트웨어 설치 | <ul style="list-style-type: none"> . 소프트웨어 설치계획 개발 . 소프트웨어 설치 실시 |
| | 소프트웨어 수락지원 | <ul style="list-style-type: none"> . 수락 검토 및 시험 지원 . 소프트웨어 제품 인도 . 획득자 교육훈련과 지원 제공 |

정통부 고시 : 제 2004-8 호 (Cont.)

데이터 기능의 복잡도 및 기능점수 가중치

▶ 내부 논리파일

| 레코드요소 유형의 개수 | 데이터요소유형의 개수 | | |
|-----------------|-------------|-------|------|
| | 1~19 | 20~50 | 51이상 |
| 1 | 낮음 | 낮음 | 보통 |
| 2~5 | 낮음 | 보통 | 높음 |
| 6이상 | 보통 | 높음 | 높음 |

| 복잡도 | 가중치 |
|-----|-----|
| 낮음 | 7 |
| 보통 | 10 |
| 높음 | 15 |

▶ 외부 논리파일

| 레코드요소 유형의 개수 | 데이터요소유형의 개수 | | |
|--------------------|-------------|-------|------|
| | 1~19 | 20~50 | 51이상 |
| 1 | 낮음 | 낮음 | 보통 |
| 2~5 | 낮음 | 보통 | 높음 |
| 6이상 | 보통 | 높음 | 높음 |

| 복잡도 | 가중치 |
|-----|-----|
| 낮음 | 5 |
| 보통 | 7 |
| 높음 | 10 |

정통부 고시 : 제 2004-8 호 (Cont.)

트랜잭션 기능의 복잡도 및 기능점수 가중치 (1)

▶ 외부 입력

| 참조 파일 유형의 개수 | 데이터요소유형의 개수 | | |
|-----------------|-------------|------|------|
| | 1~4 | 5~15 | 16이상 |
| 0 ~ 1 | 낮음 | 낮음 | 보통 |
| 2 | 낮음 | 보통 | 높음 |
| 3이상 | 보통 | 높음 | 높음 |

| 복잡도 | 가중치 |
|-----|-----|
| 낮음 | 3 |
| 보통 | 4 |
| 높음 | 6 |

▶ 외부 출력

| 참조 파일 유형의 개 수 | 데이터요소유형의 개수 | | |
|---------------------|-------------|------|------|
| | 1~5 | 6~19 | 20이상 |
| 0~1 | 낮음 | 낮음 | 보통 |
| 2~3 | 낮음 | 보통 | 높음 |
| 4이상 | 보통 | 높음 | 높음 |

| 복잡도 | 가중치 |
|-----|-----|
| 낮음 | 4 |
| 보통 | 5 |
| 높음 | 7 |

정통부 고시 : 제 2004-8 호 (Cont.)

트랜잭션 기능의 복잡도 및 기능점수 가중치 (2)

▶ 외부 조회

| 참조 파일 유형의 개수 | 데이터요소유형의 개수 | | |
|-----------------|-------------|------|------|
| | 1~4 | 5~15 | 16이상 |
| 0 ~ 1 | 낮음 | 낮음 | 보통 |
| 2 | 낮음 | 보통 | 높음 |
| 3이상 | 보통 | 높음 | 높음 |

| 복잡도 | 가중치 |
|-----|-----|
| 낮음 | 3 |
| 보통 | 4 |
| 높음 | 6 |

정통부 고시 : 제 2004-8 호 (Cont.)

평균 복잡도 및 가중치

| 유형 | 내부논리파일 | 외부연계파일 | 외부입력 | 외부출력 | 외부조회 |
|-----|--------|--------|------|------|------|
| 가중치 | 7.5 | 5.3 | 3.9 | 5.0 | 3.8 |

단계별 기능점수당 단가

| 단계 | 분석 | 설계 | 구현 | 시험 | 계 |
|----------|--------|---------|---------|---------|---------|
| 기능점수당 단가 | 86,974 | 111,156 | 146,491 | 116,460 | 461,081 |

단계별 코드 라인당 단가

| 단계 | 분석 | 설계 | 구현 | 시험 | 계 |
|----------|---------|---------|---------|---------|---------|
| 코드라인당 단가 | 1,841.1 | 2,353.0 | 3,100.9 | 2,465.2 | 9,760.2 |

정통부 고시 : 제 2004-8 호 (Cont.)

규모별 보정계수

| 스텝수 | 보정계수 | 스텝수 | 보정계수 |
|---------|------|-----------|------|
| 10,000 | 0.65 | 300,000 | 1.11 |
| 30,000 | 0.85 | 500,000 | 1.17 |
| 70,000 | 0.97 | 700,000 | 1.21 |
| 150,000 | 1.05 | 1,000,000 | 1.24 |

개발언어별 보정계수

| 개발언어 | 보정계수 |
|---|------|
| EXCEL, Spread sheet, Screen painter 등 | 0.6 |
| Delphi, HTML, SQL, Visual Basic, Small talk 등 | 0.8 |
| COBOL, Fortran, PL/I, PASCAL, Ada | 1.0 |
| C, C++, Java, CHILL, PROLOG, Shell Scripts | 1.2 |
| Assembly, 기계어, 자연어 | 1.9 |

정통부 고시 : 제 2004-8 호 (Cont.)

적용대상 기종별 보정계수

| 기종 | 보정계수 |
|------------------------|------|
| PC | 0.8 |
| Workstation (CS, Unix) | 1.1 |
| 중대형 | 1.0 |

프로젝트 규모별 보정계수

| 프로젝트 형태 | 보정계수 | 프로젝트 형태 | 보정계수 |
|--------------|------|--------------|------|
| 업무처리용 소프트웨어 | 1.0 | 시스템용 소프트웨어 | 1.7 |
| 과학기술용 소프트웨어 | 1.2 | 통신용 소프트웨어 | 1.9 |
| 멀티미디어용 소프트웨어 | 1.3 | 공정제어용 소프트웨어 | 2.0 |
| 지능정보용 소프트웨어 | 1.7 | 지휘, 통제 소프트웨어 | 2.2 |

정통부 고시 : 제 2004-8 호 (Cont.)

프로젝트 형태 분류 기준(예시)

| | |
|--------|--|
| 업무처리용 | 인사, 회계, 급여, 영업, 의사결정지원 등 경영관리용 및 특정분야의 업무용 소프트웨어 등 |
| 과학기술용 | 과학계산, 시뮬레이션, 스프레드시트, 통계, OR, CAE 등 |
| 멀티미디어용 | 그래픽, 영상처리, 음성처리, 멀티미디어 응용분야, 교육, 오락용 등 |
| 지능정보용 | 자연어처리, 인공지능, 지식정보, 전문가시스템 등 |
| 시스템용 | 운영체제, 언어처리 프로그램, DBMS, 인간-기계 인터페이스, 원도시스템, CASE, 유틸리티 등 |
| 통신제어용 | 통신프로토콜, 에뮬레이션, 교환기S/W 등 |
| 공정제어용 | 생산지원, CAM, CIM, 기기제어, 로봇분야 등 |
| 지휘 통제용 | 군, 경찰 관련 장비-인력의 지휘 및 통제에 직접적으로 관련된 소프트웨어 |

정통부 고시 : 제 2004-8 호 (Cont.)

품질 및 특성 보정계수 (1/4)

| 보정요소 | | 판단기준 | 영향도 |
|------|------------------------------|--|-----|
| 분산처리 | 어플리케이션이 구성요소 간에 데이터를 전송하는 정도 | 분산처리에 대한 요구사항이 명시되지 않음 | 0 |
| | | 클라이언트/서버 및 웹 기반 어플리케이션과 같이 분산 처리와 자료 전송이 온라인으로 수행됨 | 1 |
| | | 어플리케이션상의 처리기능이 복수개의 서버 또는 프로세서상에서 동적으로 상호 수행됨 | 2 |

정통부 고시 : 제 2004-8 호 (Cont.)

품질 및 특성 보정계수 (2/4)

| 보정요소 | | 판단기준 | 영향도 |
|--------|--------------------------|--|-----|
| 성 능 | 응답시간 또는 처리율에 대한 사용자 요구수준 | 성능에 대한 특별한 요구사항이나 활동이 명시되지 않으며, 기본적인 성능이 제공됨 | 0 |
| | | 응답시간 또는 처리율이 피크타임 또는 모든 업무시간에 중요함. 연동 시스템의 처리 마감시간에 대한 제한이 있음. | 1 |
| | | 성능 요구사항을 만족하기 위해 설계 단계에서부터 성능 분석이 요구되거나, 설계·개발·구현 단계에서 성능 분석 도구가 사용됨 | 2 |

정통부 고시 : 제 2004-8 호 (Cont.)

품질 및 특성 보정계수 (3/4)

| 보정요소 | | 판단기준 | 영향도 |
|------|----------------|---|-----|
| 신뢰성 | 장애시 미치는 영향의 정도 | 신뢰성에 대한 요구사항이 명시되지 않으며, 기본적인 신뢰성이 제공됨 | 0 |
| | | 고장시 쉽게 복구가능한 수준의 약간 불편한 손실이 발생함. | 1 |
| | | 고장시 복구가 어려우며, 재정적 손실이 많이 발생하거나, 인명피해 위험이 있음 | 2 |

정통부 고시 : 제 2004-8 호 (Cont.)

품질 및 특성 보정계수 (4/4)

| 보정요소 | | 판단기준 | 영향도 |
|--------|-----------------------------|--|-----|
| 다중 사이트 | 상이한 하드웨어와 소프트웨어 환경을 지원하는 정도 | 설계 단계에서 하나의 설치 사이트에 대한 요구사항만 고려됨. 어플리케이션이 동일한 하드웨어 또는 소프트웨어 환경하에서만 운영되도록 설계됨 | 0 |
| | | 설계 단계에서 하나 이상의 설치 사이트에 대한 요구사항이 고려됨. 어플리케이션이 유사한 하드웨어 또는 소프트웨어 환경하에서만 운영되도록 설계됨 | 1 |
| | | 설계 단계에서 하나 이상의 설치 사이트에 대한 요구사항이 고려됨. 어플리케이션이 상이한 하드웨어 및 소프트웨어 환경하에서 동작하도록 설계됨 | 2 |

정통부 고시 : 제 2004-8 호 (Cont.)

품질 및 특성 보정계수

$$= 0.025 * \text{총 영향도} + 1$$

총 영향도

$$= \text{분산처리 영향도} + \text{성능 영향도} + \text{신뢰성 영향도} + \\ \text{다중사이트 영향도}$$

Project Control

Monitor the progress of the activities.

Detect when the deviations from the plan are occurring.

Use project control techniques:

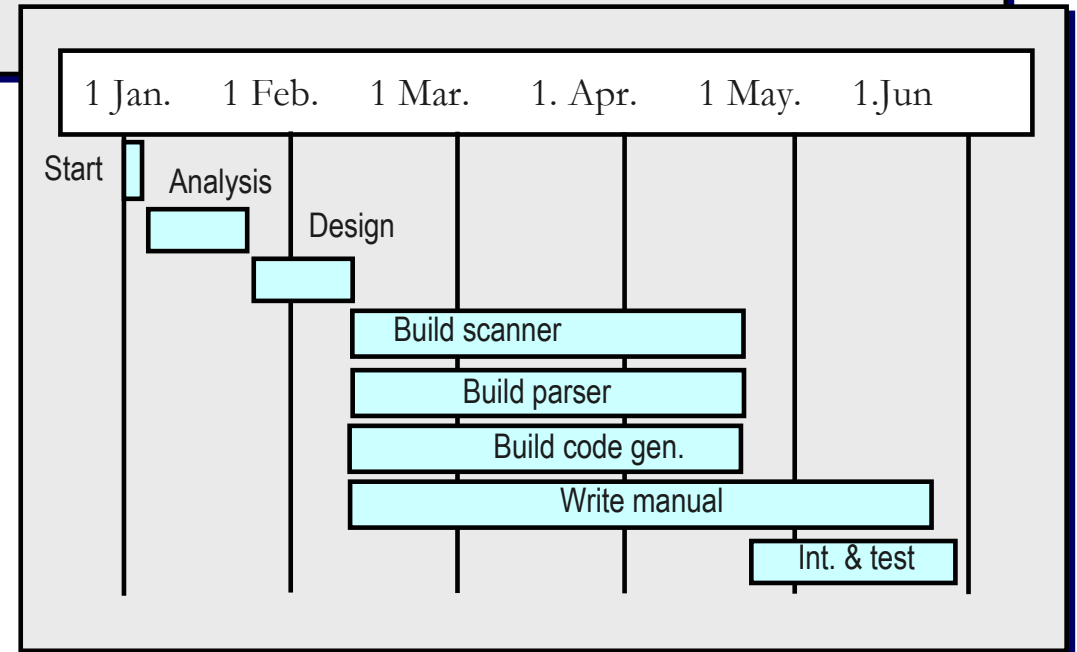
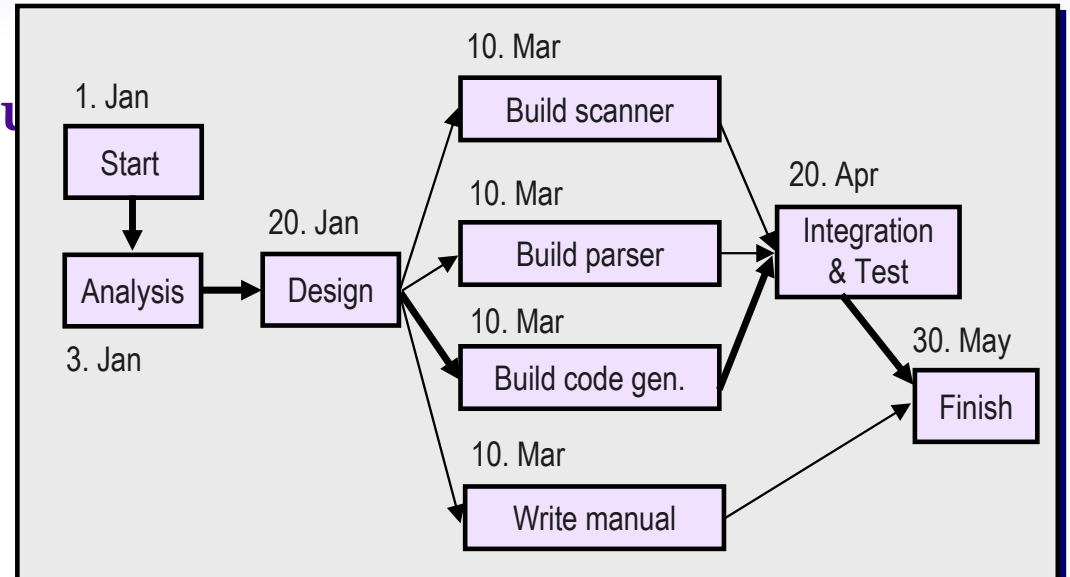
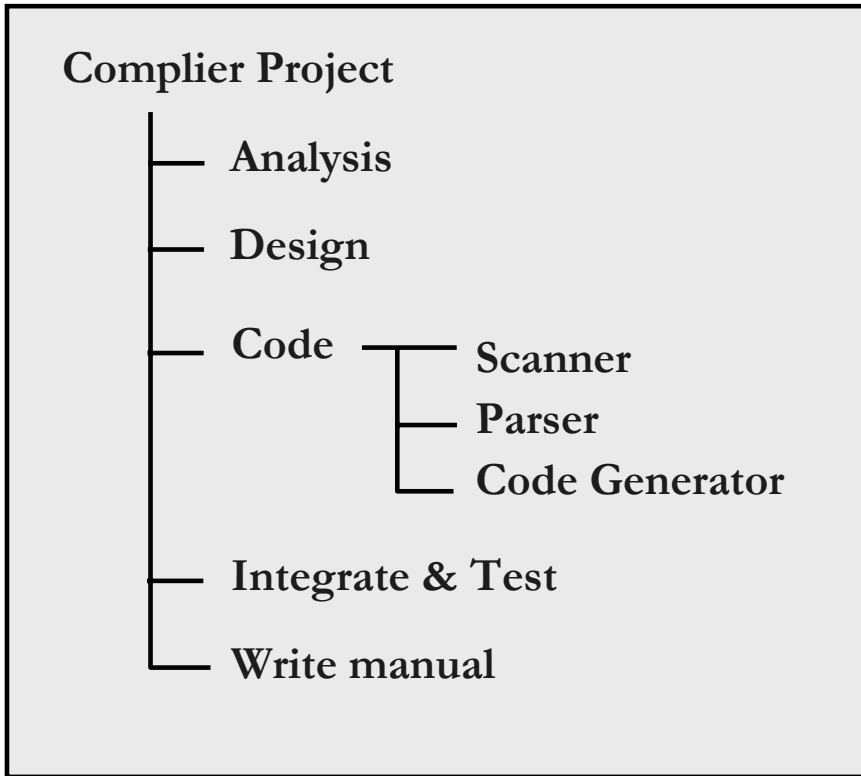
- work breakdown structure
- Gantt chart
- PERT chart(program evaluation & review technique)

Dealing with the deviations from the plan:

- adding more people may not help.
- very few options available for software projects.
- Temporarily assigning senior engineers to a part of the project.
- Remove not absolutely necessary requirements.
- Admit the incorrectness of the original plans and schedules, and revise the schedule.

Project Control Techniques

WBS (Work Breakdown Structure)
 Gantt chart
 PERT chart



Project Control Techniques – WBS (1)

WBS (Work Breakdown Structure)

- Based on breaking down the goal of the project into several intermediate goals
- WBS Goal
 - Identify all the activities that a project must undertake
- A tree whose root is labeled by the major activity of the project
 - Broken down into smaller component
 - Until each leaf in the tree represents a piece of work that the manager feels confident to estimate in term of size, difficulty, and resource requirements
- Used for
 - Summary of the project plans
 - Input into the scheduling process

Project Control Techniques – WBS (2)

An Example of WBS

| ID | 작업 아이콘 | 작업 % | 작업 이름 | 기간 | 시작 날짜 | 완료 날짜 | 선행 작업 | 자원 이름 |
|----|--------|------|----------------|-------|--------------|--------------|-------|-------|
| 1 | | 0% | 은행 SPI Project | 424 일 | 04-01-05 (월) | 05-08-29 (월) | | |
| 2 | | 0% | 현황진단단계 | 17 일 | 04-01-05 (월) | 04-01-30 (금) | | |
| 3 | | 0% | SPI TFT 구성 | 1 일 | 04-01-05 (월) | 04-01-05 (월) | | |
| 4 | | 0% | SPI 운영위원회 구성 | 3 일 | 04-01-06 (화) | 04-01-08 (목) | 3 | |
| 5 | | 0% | SPI 세미나 | 2 일 | 04-01-06 (화) | 04-01-07 (수) | 3 | |
| 6 | | 0% | 현황진단 | 9 일 | 04-01-08 (목) | 04-01-20 (화) | 3,7 | |
| 7 | | 0% | 프로세스 교육 | 2 일 | 04-01-06 (화) | 04-01-07 (수) | 3 | |
| 8 | | 0% | PIT 구성 | 4 일 | 04-01-09 (금) | 04-01-14 (수) | 4 | |
| 9 | | 0% | 우선순위 결정 | 1 일 | 04-01-26 (월) | 04-01-26 (월) | 6 | |
| 10 | | 0% | SPI 계획 수립 및 승인 | 5 일 | 04-01-26 (월) | 04-01-30 (금) | 6 | |
| 11 | | 0% | SPI TFT 멘터링 | 15 일 | 04-01-06 (화) | 04-01-29 (목) | 3 | |
| 12 | | 0% | 단계 종료 보고 | 1 일 | 04-01-30 (금) | 04-01-30 (금) | 11 | |
| 13 | | 0% | 구축 및 초기적용 단계 | 126 일 | 04-02-02 (월) | 04-07-30 (금) | 2 | |
| 14 | | 0% | 표준 프로세스 개발 | 70 일 | 04-02-02 (월) | 04-05-12 (수) | | |
| 15 | | 0% | 프로세스 개발 | 40 일 | 04-02-02 (월) | 04-03-29 (월) | | |
| 16 | | 0% | 템플릿 개발 | 12 일 | 04-03-30 (화) | 04-04-15 (목) | 15 | |
| 17 | | 0% | 가이드라인 개발 | 12 일 | 04-04-16 (금) | 04-05-03 (월) | 16 | |
| 18 | | 0% | 체크리스트 개발 | 6 일 | 04-05-04 (화) | 04-05-12 (수) | 17 | |
| 19 | | 0% | 방법론 개발 | 50 일 | 04-02-02 (월) | 04-04-13 (화) | 2 | |
| 20 | | 0% | 객체지향 방법론 | 25 일 | 04-02-02 (월) | 04-03-08 (월) | | |
| 21 | | 0% | 정보공학 방법론 | 25 일 | 04-03-09 (화) | 04-04-13 (화) | 20 | |
| 22 | | 0% | SPI 교육 | 12 일 | 04-05-13 (목) | 04-05-31 (월) | 14,19 | |
| 23 | | 0% | 방법론 교육 | 4 일 | 04-05-13 (목) | 04-05-18 (화) | 14,19 | |
| 24 | | 0% | 프로세스 파일럿 적용 | 43 일 | 04-06-01 (화) | 04-07-29 (목) | 22,23 | |
| 25 | | 0% | 프로젝트 멘터링 및 코칭 | 43 일 | 04-06-01 (화) | 04-07-29 (목) | 22,23 | |
| 26 | | 0% | 단계 종료 보고 | 1 일 | 04-07-30 (금) | 04-07-30 (금) | 24,25 | |

Project Control Techniques – Gantt chart

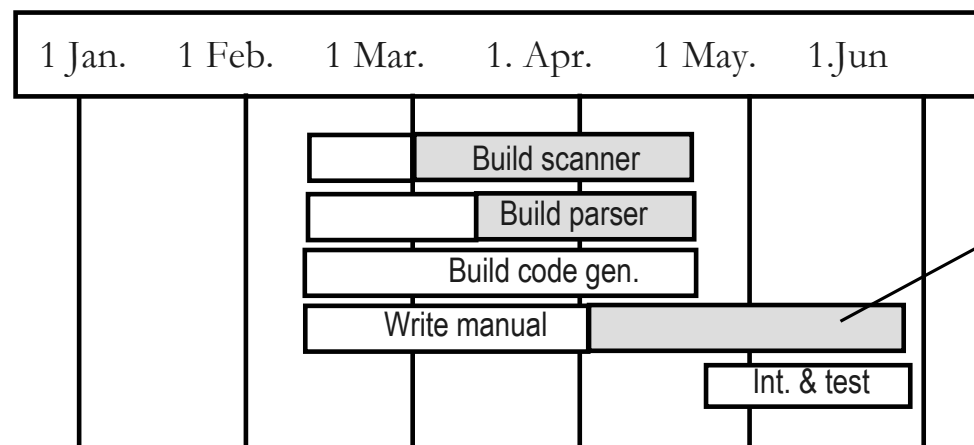
A techniques that can be used for scheduling, budgeting, and resource planning.

A kind of bar chart

- Each bar represents an activity
- Drawn against a timeline (proportional to the length of time planned for the activity)

Used to allocate resources and plan staffing

Do not highlight inter-task dependencies



Slack time : the latest time by which a task must be finished

Project Control Techniques – PERT chart

Program Evaluation and Review Techniques

Represents the dependencies of activities which should performed in a project

- A network of boxes (activities) and arrows (dependencies of activities)

Dates from PERT chart

- Earliest start date and latest start date
- Earliest finish date and latest finish date

Critical path

- A path that cause a delay in the entire project because of any delay in any activity (ref. bold-line path of PERT chart at slide # 21)

Salient features

- Expose all possible parallelism in the activities
- Allows scheduling and simulation of alternative schedules
- Enables the manager to monitor and control the project

Summary and Discussion

소프트웨어의 사이즈 및 비용 예측 방법들

- SLOC : Source Line of Code
- FP : Function Point

COCOMO (II)

- Constructive Cost Model

소프트웨어사업대가의 기준 : 정통부 고시 제2004-8호

(?) 소프트웨어 사이즈 및 비용을 보다 정확하게 예측하려는 이유는 무엇인가?





프로젝트 관리론

- 기능점수에 의한 비용예측 (2) -

2005. 10

홍 장 의

jehong@chungbuk.ac.kr

목 차

I. 기능점수(Function Point) 개요

II. Function Point 분석 방법

III. 프로젝트 타입별 Function Point 산출

IV. Function Point 관련 척도 및 이득

V. 결론

첨부 : 약어

기능점수(Function Point)

개 요

Function Point

■ Function Point 정의

- 소프트웨어 시스템에 대한 산출물 사이즈의 측정 단위
- 소프트웨어 시스템의 기능적 측면의 측정
 - 데이터 기능 (Data Function)
 - 트랜잭션 기능 (Transaction Function)

■ Function Point 역사

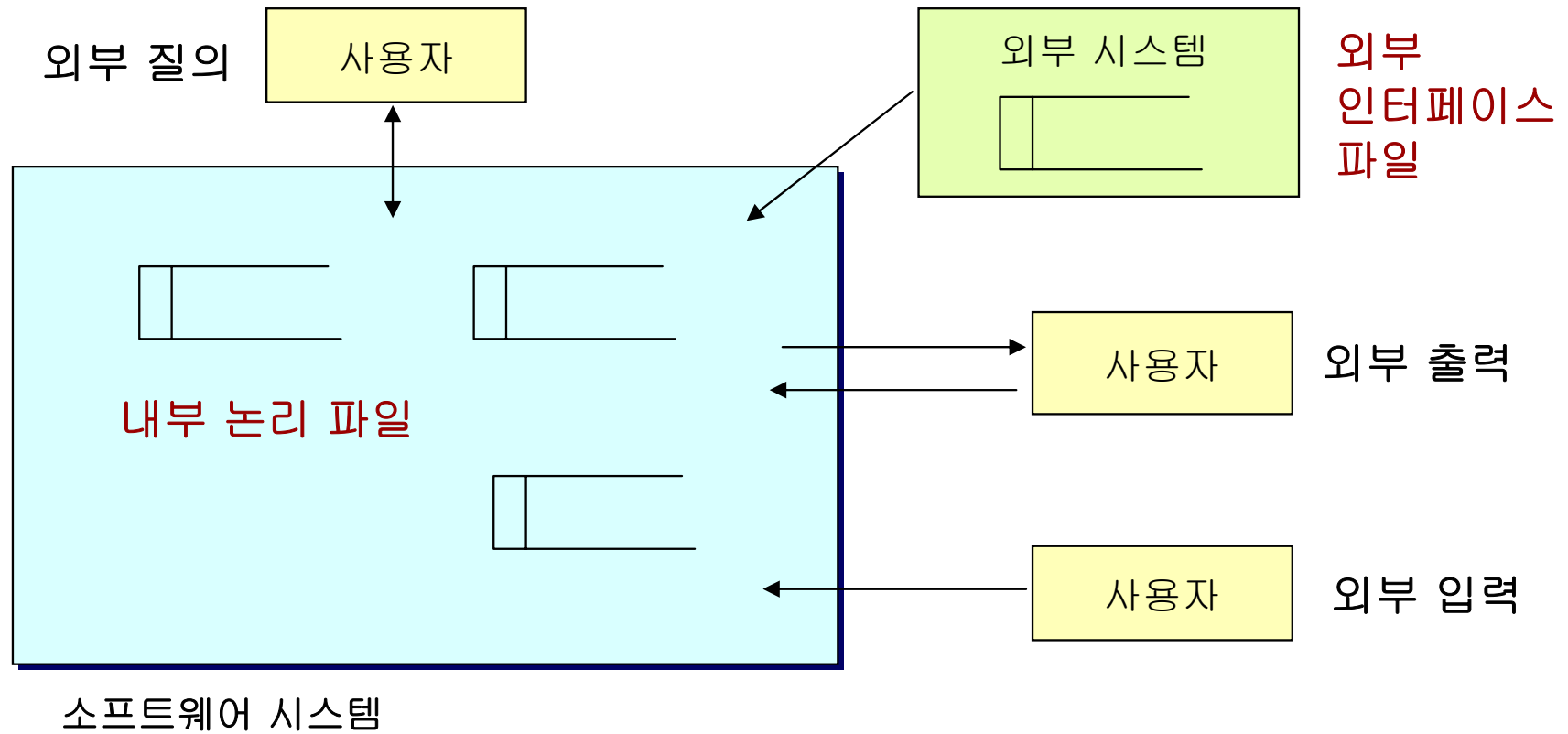
- 1979 : Allan Albrecht, IBM
- 1986 : IFPUG(International Function Point User group)
- 1987 : British Government, Standard S/W Productivity Metric
- 1990 : IFPUG, Function Point Counting Practices Manual, Rel. 3.0
- 1994 : IFPUG Release 4.0
- 1999 : IFPUG Release 4.1, Currently

Function Point

■ Function Point 역사

- 1979 : Allan Albrecht, IBM
- 1986 : IFPUG(International Function Point User group)
- 1987 : British Government, Standard S/W Productivity Metric
- 1990 : IFPUG, Function Point Counting Practices Manual, Rel. 3.0
- 1994 : IFPUG Release 4.0
- 1998 : ISO/IEC 14143-1 : Functional Size Measurement (FSM) - Concept
- 1999 : IFPUG Release 4.1
- 2002 : ISO/IEC 14143-2/3/4 : FSM – Conformance Evaluation/Verification/Reference Model
- 2003 : ISO/IEC 20926 IFPUG 4.1 Unadjusted functional size measurement method - Counting practices manual

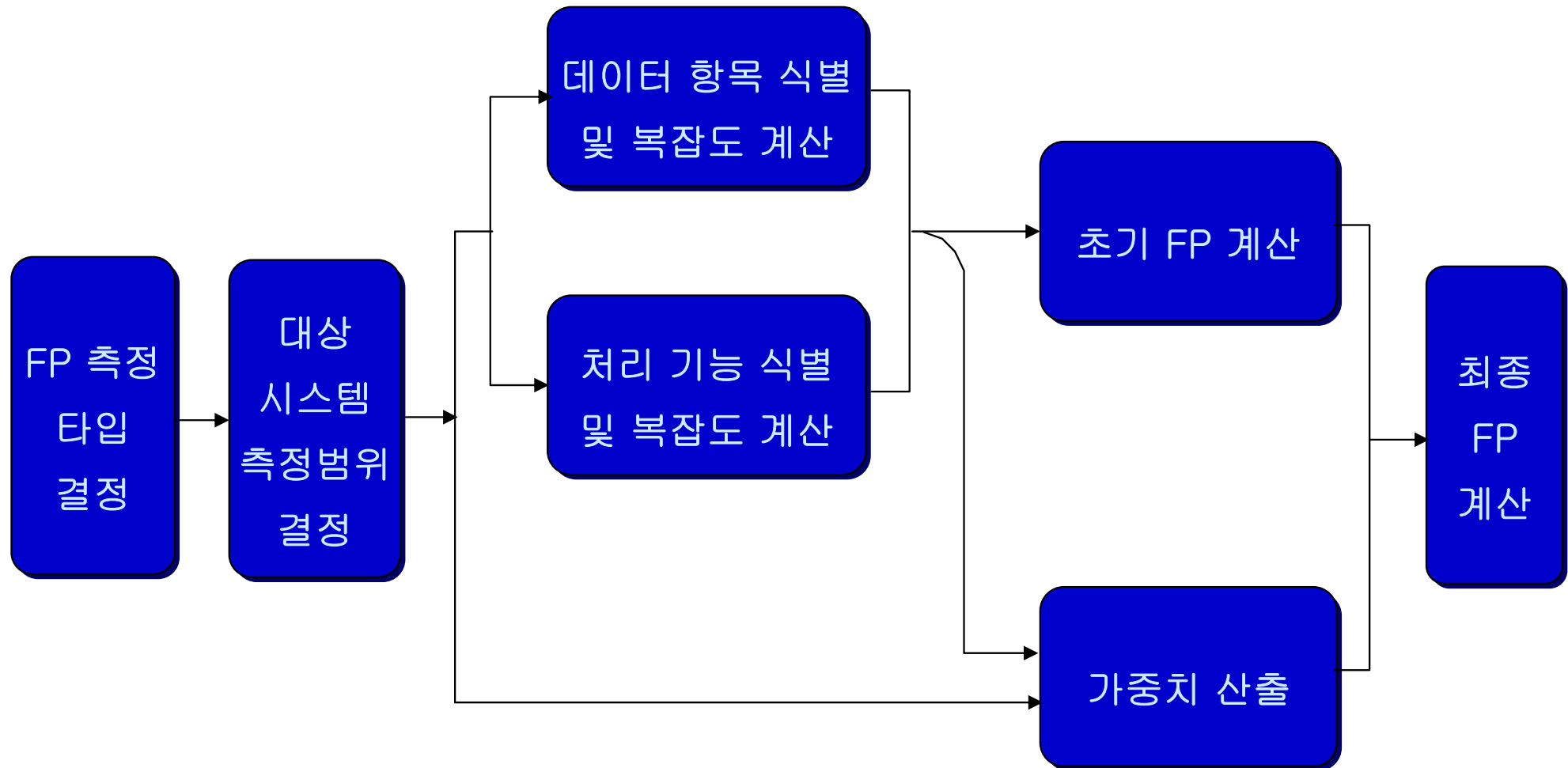
일반적인 소프트웨어 시스템의 기능 특성



- 데이터 기능 : 내부 논리 파일 + 외부 인터페이스 파일
- 트랜잭션 기능 : 외부 질의 + 외부 출력 + 외부 입력

Function Point 분석 방법

분석 절차도



단계 1: FP 측정 타입 결정

■ FP 타입

- 측정대상이 되는 산출물이 다음의 3가지 프로젝트 유형 중에서는 어느 타입의 프로젝트로부터 산출되는가를 결정
 - 신규 개발 프로젝트 타입
 - 개선 프로젝트 타입
 - 유지보수 프로젝트 타입



단계 2 : 대상 시스템 측정 범위 결정

2.1 측정 대상 범위 결정

- 프로젝트 개발 기능 전체
- 핵심 기능 일부

2.2 측정 대상 결정

- 자체 개발 부분
- 외부 개발 부분
- 구매 패키지 부분
- 기타 부분

2.3 대상 시스템 측정 한계도

- 블록 다이어그램
- 개체 관계도
- 시스템 문맥도
- 상위 유즈케이스 다이어그램

양식 1. FP 분석 및 산출을 위한 프로젝트 요약

| | | |
|-----------------------|---|---|
| 프로젝트 명 | | |
| 프로젝트 개요 | | |
| 프로젝트 타입 | <input type="checkbox"/> 신규 개발 프로젝트 <input type="checkbox"/> 개선 프로젝트 <input type="checkbox"/> 유지보수 프로젝트 | |
| 프로젝트 주무부서 | | |
| 프로젝트 범위 및 FP 측정 범위 | 측정 범위 | <input type="checkbox"/> 프로젝트 전체 <input type="checkbox"/> 시스템 일부 <input type="checkbox"/> 기타 : |
| | 측정 대상 | <input type="checkbox"/> 자체 개발 부분 <input type="checkbox"/> 외주 개발 부분 <input type="checkbox"/> 구매 패키지 부분 <input type="checkbox"/> 기타 : |
| FP 분석 참여자 | 1. 분석가 : 3. 분석가 : | 2. 분석가 : 4. 조정자 : |
| 참고 사항 | | |

양식 2. 시스템 구성도 및 FP 측정 범위

| | |
|------------|--|
| 프로젝트 명 | |
| 시스템 구성도 | |
| | |
| 인터페이스 정의사항 | |
| 참고 사항 | |

개발 시스템 요구사항 정의

■ 시스템 명 : 고객 계좌 정보관리 시스템

■ 주요 시스템 요구사항

- (1) 고객에 대한 정보를 유지관리, 조회, 출력한다.
- (2) 출력은 외부에 존재하는 고객에 대한 신용정보를 포함할 수 있다.
- (3) 은행은 자체적으로 계좌를 구성하는 다양한 상품에 대한 정보를 유지, 조회, 출력한다.
- (4) 고객이 보유하는 계좌에 대한 정보를 유지관리, 조회, 출력한다.
- (5) 은행은 신용 정보(등급)에 따른 고객의 목록을 조회하고 출력한다.

■ 주요 구성 개체

- 고객 : 은행에 등록된 모든 고객으로, 크게 개인고객과 법인고객으로 분류
- 신용정보 : 등록된 모든 고객에 대한 신용 정보 및 등급
- 계좌 : 등록된 고객이 보유한 상품에 대한 계좌번호
- 상품 : 은행에서 취급하는 모든 상품 목록
- 상품정보 : 각 상품에 대한 세부적인 정보

양식 1. FP 분석 및 산출을 위한 프로젝트 요약

| | | |
|-----------------------|--|--|
| 프로젝트 명 | 고객 계좌 정보관리 시스템 개발 | |
| 프로젝트 개요 | 본 프로젝트는 현재 본사에 등록된 모든 고객에 대한 계좌 정보를 유지 관리하기 위한 시스템 구축을 목표로 한다. | |
| 프로젝트 타입 | <input checked="" type="checkbox"/> 신규 개발 프로젝트 <input type="checkbox"/> 개선 프로젝트 <input type="checkbox"/> 유지보수 프로젝트 | |
| 프로젝트 주무부서 | 계정계 | |
| 프로젝트 범위 및 FP 측정 범위 | 측정 범위 | <input checked="" type="checkbox"/> 프로젝트 전체 <input type="checkbox"/> 시스템 일부 <input type="checkbox"/> 기타 : |
| | 측정 대상 | <input checked="" type="checkbox"/> 자체 개발 부분 <input type="checkbox"/> 외주 개발 부분 <input type="checkbox"/> 구매 패키지 부분 <input type="checkbox"/> 기타 : |
| FP 분석 참여자 | 1. 분석가 : 김 유 신 2. 분석가 : 이 순 신 3. 분석가 : 변 사 또 4. 조정자 : 최 무 선 | |
| 참고 사항 | - 개발 환경은 윈도우 환경에서의 GUI를 채택한다. | |

양식 2. 시스템 구성도 및 FP 측정 범위

| | |
|--|---|
| 프로젝트 명 | 고객 계좌 정보관리 시스템 개발 |
| 시스템 구성도 | |
| <pre> classDiagram class Customer { Individual Corporation } class Account class Credit class Product class Prod_Desc Customer < -- Account Customer < -- Credit Account < -- Product Product < -- Prod_Desc </pre> | |
| 인터페이스 정의사항 | 고객에 대한 신용정보(Credit)는 외부에 존재하는 종합적인 개인 신용 평가에 의거한다 |
| 참고 사항 | |

단계 3 : 데이터 기능 식별 및 복잡도 계산

3.1 내부 논리 파일(ILF) 식별

3.2 외부 인터페이스 파일(EIF) 식별

3.3 ILF 및 EIF 복잡도 산출

내부 논리 파일

- Internal Logical File
- 응용 시스템 범주 내에서 유지 관리 되는 파일
- 사용자에 의해 식별 가능한 데이터 및 데이터 그룹
- 논리적 관련성

절차

3.1.1 ILF 식별

3.1.2 ILF 구성 데이터 정의

단계 3 : 데이터 기능 식별 및 복잡도 계산 (계속)

3.1 내부 논리 파일(ILF)
식별

3.2 외부 인터페이스
파일(EIF) 식별

3.3 ILF 및 EIF 복잡도
산출

외부 인터페이스 파일

- External Interface File
- 응용 시스템에서 다른 응용 시스템에서 유지하는 파일을 참조
- 사용자에게 의해 식별 가능한 데이터 및 데이터 그룹
- 논리적 관련성

절차

3.2.1 EIF 식별

3.2.2 EIF 구성 데이터 정의

양식 3. ILF 및 EIF 파일 정의

| 관련 기능명 | 파일/개체명 | 파일 타입 | 필드명 | 필드타입 | 비고 | |
|--------|-------------|-------|-------------------|--------------|-----------------------------|--|
| 고객 정보 | Customer | | C_Name | | C_Name, C_IDN은 주키로 사용 | |
| | | | C_IDN (SSN / CRN) | | | |
| | | | C_Phone | | | |
| | | | C_Address | | | |
| | | | C_Type_Code | | | |
| | | | Credit_Grade (FK) | | | |
| | Individual | | | Job_Name | | |
| | | | | Year_Income | | |
| | | | | No_Family | | |
| | Corporation | | | Total_Fund | | |
| | | | | Service_Type | | |
| | | | | No_Employee | | |
| 신용 등급 | Credit | | Credit_Grade | | | |
| | | | Grade_Name | | | |
| | | | No_Shift | | | |
| | | | Constraints | | | |

양식 3. ILF 및 EIF 파일 정의 (계속)

| 관련 기능명 | 파일/개체명 | 파일 타입 | 필드명 | 필드타입 | 비고 | |
|--------|-----------|-------|------------------|----------------|----|---|
| 계좌 정보 | Account | | Account_No | | | |
| | | | PIN_No | | | |
| | | | Open_Date | | | |
| | | | Target_Amount | | | |
| | | | Target_Period | | | |
| | | | Acc_Constraints | | | |
| | | | C_IDN (FK) | | | |
| | | | Product_UID (FK) | | | |
| 상품 정보 | Product | | Product_UID | | | |
| | | | P_Type | | | |
| | | | P_Name | | | |
| | | | Subsc_Period | | | |
| | Prod_Desc | | | Product_UID | | Product와 한 파일이나 기술적인 이유 로 별도유지 |
| | | | | Prod_Agreement | | |
| | | | | Line_Number | | |
| | | | | Number_Desc | | |

단계 3 : 데이터 기능 식별 및 복잡도 계산 (계속)

3.1 내부 논리 파일(ILF)
식별

3.2 외부 인터페이스
파일(EIF) 식별

3.3 ILF 및 EIF 복잡도
산출

복잡도 산출

- 각 파일의 기능 복잡도
- 상,중,하로 구분하여 식별

절차

3.3.1 데이터 타입 결정

- 데이터 요소 타입
- 레코드 요소 타입

3.3.2 기능 복잡도 산출

- 데이터 요소 타입의 수
- 레코드 요소 타입의 수

단계 3 : 데이터 기능 식별 및 복잡도 계산 (계속)

■ ILF 및 TIF에 대한 기능 복잡도 메트릭

| | | 데이터 요소 타입 | | |
|-----------------|-------|-----------|---------|-------|
| | | 1 - 19 | 20 - 50 | >= 51 |
| 레코드 요소 타입 | 1 | 하 | 하 | 중 |
| | 2 - 5 | 하 | 중 | 상 |
| | > 5 | 중 | 상 | 상 |

양식 3. ILF 및 EIF 파일 정의

| 관련 기능명 | 파일/개체명 | 파일 타입 | 필드명 | 필드타입 | 비고 |
|-------------|-------------|-----------------|-------------------|--------|-----------------------------|
| 고객 정보 | Customer | ILF | C_Name | DET_1 | C_Name, C_IDN은 주키로 사용 |
| | | | C_IDN (SSN / CRN) | DET_2 | |
| | | | C_Phone | DET_3 | |
| | | | C_Address | DET_4 | |
| | | | C_Type_Code | DET_5 | |
| | | | Credit_Grade (FK) | DET_6 | |
| | Individual | SubFile RET | Job_Name | DET_7 | |
| | | | Year_Income | DET_8 | |
| | | | No_Family | DET_9 | |
| | Corporation | SubFile RET | Total_Fund | DET_10 | |
| | | | Year_Revenue | DET_11 | |
| | | | Service_Type | DET_12 | |
| No_Employee | | | DET_13 | | |
| 신용 등급 | Credit | EIF & RET | Credit_Grade | DET_1 | |
| | | | Grade_Name | DET_2 | |
| | | | No_Shift | DET_3 | |
| | | | Constraints | DET_4 | |

양식 3. ILF 및 EIF 파일 정의 (계속)

| 관련 기능명 | 파일/개체명 | 파일 타입 | 필드명 | 필드타입 | 비고 |
|--------|-----------|-----------------|------------------|-------|---|
| 계좌 정보 | Account | ILF & RET | Account_No | DET_1 | |
| | | | PIN_No | DET_2 | |
| | | | Open_Date | DET_3 | |
| | | | Target_Amount | DET_4 | |
| | | | Target_Period | DET_5 | |
| | | | Acc_Constraints | DET_6 | |
| | | | C_IDN (FY) | DET_7 | |
| | | | Product_UID (FY) | DET_8 | |
| 상품 정보 | Product | ILF & RET | Product_UID | DET_1 | Product와 한 파일이나 기술적인 이유 로 별도유지 |
| | | | P_Type | DET_2 | |
| | | | P_Name | DET_3 | |
| | | | Subsc_Period | DET_4 | |
| | Prod_Desc | SubFile (X) | Product_UID | X | |
| | | | Prod_Agreement | DET_5 | |
| | | | Line_Number | DET_6 | |
| | | | Number_Desc | DET_7 | |

양식 8. 데이터 기능(ILF/EIF) 복잡도 계산

| 구분 | 파일/개체명 | # DET | #RET/#FTR | 복잡도 | 기능수준가중치 | 초기 FP |
|----------------------|----------|-------|-----------|-----|---------|-------|
| ILF | Customer | 13 | 2 | 하 | | |
| | Account | 8 | 1 | 하 | | |
| | Product | 7 | 1 | 하 | | |
| EIF | Credit | 4 | 1 | 하 | | |
| EI | | | | | | |
| EO | | | | | | |
| EQ | | | | | | |
| 초기 Function Point 합계 | | | | | | |

단계 4 : 트랜잭션 기능 식별 및 복잡도 계산

- 응용 시스템 내에서 데이터 처리를 통해 사용자에게 서비스 기능을 제공하는 처리
- 처리 기능의 3 유형
 - 외부 입력(External Input)
 - ILF를 유지하기 위한 입력 데이터와 시스템의 행위를 통제하기 위한 제어 정보를 처리한다.
 - 외부 질의(External inQuery)
 - ILF 또는 EIF로 부터 데이터 또는 제어 정보의 조회를 통해 응용 시스템의 외부로 데이터를 보낸다.
 - 외부 출력(External Output)
 - 데이터 또는 제어 정보의 조회 보다는 처리 로직을 통해 응용 시스템의 외부로 데이터를 보낸다.

단계 4 : 트랜잭션 기능 식별 및 복잡도 계산 (계속)

■ 절차

4.1 처리 프로세스 식별

- 4.1.1 ILF에 대한 처리 프로세스 도출
- 4.1.2 EIF에 대한 처리 프로세스 도출

4.2 외부 입력 기능 식별 및 복잡도 산출

- 4.2.1 ILF 유지관리 처리 프로세스 추출
- 4.2.2 처리 프로세스별 데이터 요소 식별
- 4.2.3 처리 프로세스별 참조 파일 식별
- 4.2.4 외부 입력의 복잡도 산출

4.3 외부 출력 기능 식별 및 복잡도 산출

- 4.3.1 외부 출력에 대한 처리 프로세스 추출
- 4.3.2 처리 프로세스별 데이터 요소 식별
- 4.3.3 처리 프로세스별 참조 파일 식별
- 4.3.4 외부 출력에 대한 복잡도 산출

4.4 외부 질의 기능 식별 및 복잡도 산출

- 4.4.1 외부 질의에 대한 처리 프로세스 추출
- 4.4.2 처리 프로세스별 데이터 요소 식별
- 4.4.3 처리 프로세스별 참조 파일 식별
- 4.4.4 외부 질의에 대한 복잡도 산출

단계 4 : 트랜잭션 기능 식별 및 복잡도 계산 (계속)

■ 4.1 처리 프로세스 식별

- 처리 프로세스에 대한 가정

| | |
|----------|--|
| DET에 대하여 | <ul style="list-style-type: none"> - 생성 및 갱신은 ILF에 존재하는 모든 필드를 접근한다. - 삭제기능은 해당 ILF의 주 키만을 삭제한다. - 계좌 정보에 생성 및 변경 사항을 반영해야 한다. - 약관의 변경에 따라 계좌의 제약사항(Constraints)가 변경되어야 한다. |
| FTR에 대하여 | <ul style="list-style-type: none"> - 신규 고객을 생성하기 위해서는 신용 정보를 참조한다. |

양식 4. ILF/EIF에 대한 처리 프로세스 정의

| 파일 타입 | 파일 명 | 처리 프로세스명 | 참고 설명 |
|----------------|----------|--------------------|-------|
| ILF | Customer | Create_Customer | |
| | | Customer_Inquiry | |
| | | Update_Customer | |
| | | Delete_Customer | |
| | | Customer_Report | |
| | Account | Assign_Account | |
| | | Account_Inquiry | |
| | | Update_Account | |
| | | Delete_Account | |
| | | Change_Constraints | |
| | | Account_Report | |
| | Product | Create_Product | |
| | | Product_Inquiry | |
| | | Update_Product | |
| | | Delete_Product | |
| Product_Report | | | |
| EIF | Credit | Credit_Inquiry | |
| | | Credit_Report | |

■ 4.2 외부 입력 기능 식별 및 복잡도 산출

- ILF에 대한 생성, 갱신, 삭제 프로세스
- Create_Customer
 - 생성과 갱신은 ILF 파일의 모든 필드를 접근
 - Customer를 생성하기 위해 Credit 정보 이용
 - 동일한 항목의 접근은 1번만 고려

ILF : Customer 필드 13개
EIF : Credit 필드 4개



Customer : 13 DETs
Credit : 3 DETs

- Delete_Customer
 - 삭제 기능은 해당 파일의 주키만 삭제
 - Customer의 삭제는 account 삭제로 연결됨

ILF : Customer의 Primary Key : 2개
ILF : Account의 Primary Key : 1개

양식 5. 외부 입력(EI) 트랜잭션에 대한 정의

| ILF 파일 명 | 처리 프로세스 | 데이터 요소 타입 | 참조 파일 |
|----------|--------------------|---------------------|----------------------------|
| Customer | Create_Customer | 13 DETs of Customer | Customer Credit |
| | | 3 DETs of Credit | |
| | Update_Customer | 13 DETs of Customer | Customer Credit |
| | | 3 DETs of Credit | |
| | Delete_Customer | 2 DETs of Customer | Customer Account |
| | | 1 DETs of Account | |
| Account | Assign_Account | 8 DETs of Account | Account, Customer, Product |
| | Update_Account | 8 DETs of Account | Account, Customer, Product |
| | Change_Constraints | 8 DETs of Account | Account |
| | Delete_Account | 1 DETs of Account | Account |
| Product | Create_Product | 7 DETs of Product | Product |
| | Update_Product | 7 DETs of Product | Product |
| | Delete_Product | 2 DETs of Product | Product Account |
| | | 1 DETs of Account | |

단계 4 : 트랜잭션 기능 식별 및 복잡도 계산 (계속)

■ 외부 입력(EI)에 대한 기능 복잡도 메트릭

| | | 데이터 요소 타입 | | |
|-------------|-------|-----------|--------|-----------|
| | | 1 - 4 | 5 - 15 | ≥ 16 |
| 참조 파일 타입 | < 2 | 하 | 하 | 중 |
| | 2 | 하 | 중 | 상 |
| | > 2 | 중 | 상 | 상 |

■ 4.3 외부 출력 기능 식별 및 복잡도 산출

- 외부 출력 처리 프로세스에 대한 가정

외부 출력
처리 프로세스에
대한 가정

- 보고서의 생성시 기존 저장된 파일의 데이터를 출력하는 경우가 일반적이다.
- 기존 데이터를 이용하여 계산 결과 및 파생 데이터를 생성하여 출력하는 항목은 유지되는 데이터에 대하여 40%정도의 신규 항목이 생성된다.
- 처리 과정에 대한 메시지나 오류 메시지는 앞의 40%에 포함된다고 가정한다.

양식 6. 외부 출력(EO) 트랜잭션에 대한 정의

| 파일 명 | 처리 프로세스 | 데이터 요소 타입 | 참조 파일 |
|----------|-----------------|---------------------|--------------------------------|
| Customer | Customer_Report | 13 DETs of Customer | Customer Credit |
| | | 3 DETs of Credit | |
| | | 6 DETs of Derived | |
| Account | Account_Report | 8 DETs of Account | Account Customer Product |
| | | 13 DETs of Customer | |
| | | 7 DETs of Product | |
| | | 11 DETs of Derived | |
| Product | Product_Report | 7 DETs of Product | Product |
| Credit | Credit_Report | 2 DETs of Credit | Credit Customer, |
| | | 5 DETs of Customer | |
| | | 3 DETs of Derived | |

단계 4 : 트랜잭션 기능 식별 및 복잡도 계산 (계속)

■ 외부 출력(EO)에 대한 기능 복잡도 메트릭

| | | 데이터 요소 타입 | | |
|----------------|-------|-----------|--------|-----------|
| | | 1 - 5 | 6 - 19 | ≥ 20 |
| 참조 파일 타입 | < 2 | 하 | 하 | 중 |
| | 2 - 3 | 하 | 중 | 상 |
| | > 3 | 중 | 상 | 상 |

■ 4.4 외부 질의 기능 식별 및 복잡도 산출

- 외부 질의 처리 프로세스에 대한 가정

외부 질의
처리 프로세스에
대한 가정

- 질의를 위해서는 해당 파일의 데이터 요소를 모두 접근한다.
- 질의를 위하여 요구되는 메시지나 명령 키 등도 포함하여 산출한다.
- 질의에 대하여 ILF로 부터 접근하는 다른 외부 파일은 없다고 가정한다.

양식 7. 외부 질의(EQ) 트랜잭션에 대한 정의

| 파일 명 | 처리 프로세스 | 데이터 요소 타입 | 참조 파일 |
|----------|------------------|---------------------|----------|
| Customer | Customer_Inquiry | 15 DETs of Customer | Customer |
| Account | Account_Inquiry | 10 DETs of Credit | Account |
| Product | Product_Inquiry | 9 DETs of Derived | Product |
| Credit | Credit_Inquiry | 6 DETs of Account | Credit |

단계 4 : 트랜잭션 기능 식별 및 복잡도 계산 (계속)

■ 외부 질의(EQ)에 대한 기능 복잡도 메트릭

| | | 데이터 요소 타입 | | |
|----------------|-------|-----------|--------|-------|
| | | 1 - 5 | 6 - 19 | >= 20 |
| 참조 파일 타입 | 1 | 하 | 하 | 중 |
| | 2 - 3 | 하 | 중 | 상 |
| | > 3 | 중 | 상 | 상 |

양식 8. 트랜잭션 기능(EI/EO/EQ) 복잡도 계산

| 구분 | 처리 프로세스명 | # DET | #RET/#FTR | 복잡도 | 기능수준가중치 | 초기 FP |
|----------------------|--------------------|-------|-----------|-----|---------|-------|
| EI | Create_Customer | 16 | 2 | 상 | | |
| | Update_Customer | 16 | 2 | 상 | | |
| | Delete_Customer | 3 | 2 | 하 | | |
| | Assign_Account | 8 | 3 | 중 | | |
| | Update_Account | 8 | 3 | 중 | | |
| | Change_Constraints | 8 | 1 | 하 | | |
| | Delete_Account | 1 | 1 | 하 | | |
| | Create_Product | 7 | 1 | 하 | | |
| | Update_Product | 7 | 1 | 하 | | |
| | Delete_Product | 3 | 2 | 하 | | |
| 초기 Function Point 합계 | | | | | | |

양식 8. 트랜잭션 기능(EI/EO/EQ) 복잡도 계산 (계속)

| 구분 | 처리 프로세스명 | # DET | #RET/#FTR | 복잡도 | 기능수준가중치 | 초기 FP |
|----------------------|------------------|-------|-----------|-----|---------|-------|
| EO | Customer_Report | 22 | 2 | 상 | | |
| | Account_Report | 39 | 3 | 상 | | |
| | Product_Report | 7 | 1 | 하 | | |
| | Credit_Report | 10 | 2 | 중 | | |
| EQ | Customer_Inquiry | 15 | 1 | 하 | | |
| | Account_Inquiry | 10 | 1 | 하 | | |
| | Product_Inquiry | 9 | 1 | 하 | | |
| | Credit_Inquiry | 6 | 1 | 하 | | |
| 초기 Function Point 합계 | | | | | | |

단계 5 : 초기 FP (UFP) 산출

- 데이터 기능과 처리 기능에 대한 복잡도 고려
 - ILF, EIF, EI, EO, 그리고 EQ
 - UFP : Unadjusted Function Point

- 기능 복잡도 수준별 가중치

| 구성 요소 | 기능 수준 | | |
|-------|--------|------------|---------|
| | 하(Low) | 중(Average) | 고(High) |
| ILF | X 7 | X 10 | X 15 |
| EIF | X 5 | X 7 | X 10 |
| EI | X 3 | X 4 | X 6 |
| EO | X 4 | X 5 | X 7 |
| EQ | X 3 | X 4 | X 6 |

양식 8. ILF/EIF/EI/EO/EQ의 초기 FP 산출

| 구분 | 파일/개체명 | # DET | #RET/#FTR | 복잡도 | 기능수준가중치 | 초기 FP |
|----------------------|----------|-------|-----------|-----|---------|-------|
| ILF | Customer | 13 | 2 | 하 | 7 | 21 |
| | Account | 8 | 1 | 하 | 7 | |
| | Product | 7 | 1 | 하 | 7 | |
| EIF | Credit | 4 | 1 | 하 | 5 | 5 |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| 초기 Function Point 소계 | | | | | | 26 |

양식 8. ILF/EIF/EI/EO/EQ의 초기 FP 산출 (계속)

| 구분 | 처리 프로세스명 | # DET | #RET/#FTR | 복잡도 | 기능수준가중치 | 초기 FP |
|----------------------|--------------------|-------|-----------|-----|---------|-------|
| EI | Create_Customer | 16 | 2 | 상 | 6 | 38 |
| | Update_Customer | 16 | 2 | 상 | 6 | |
| | Delete_Customer | 3 | 2 | 하 | 3 | |
| | Assign_Account | 8 | 3 | 중 | 4 | |
| | Update_Account | 8 | 3 | 중 | 4 | |
| | Change_Constraints | 8 | 1 | 하 | 3 | |
| | Delete_Account | 1 | 1 | 하 | 3 | |
| | Create_Product | 7 | 1 | 하 | 3 | |
| | Update_Product | 7 | 1 | 하 | 3 | |
| | Delete_Product | 3 | 2 | 하 | 3 | |
| 초기 Function Point 소계 | | | | | | 38 |

양식 8. ILF/EIF/EI/EO/EQ의 초기 FP 산출 (계속)

| 구분 | 처리 프로세스명 | # DET | #RET/#FTR | 복잡도 | 기능수준가중치 | 초기 FP |
|----------------------|------------------|-----------------------|-----------|-----|---------|-------|
| EO | Customer_Report | 22 | 2 | 상 | 7 | 23 |
| | Account_Report | 39 | 3 | 상 | 7 | |
| | Product_Report | 7 | 1 | 하 | 4 | |
| | Credit_Report | 10 | 2 | 중 | 5 | |
| EQ | Customer_Inquiry | 15 | 1 | 하 | 3 | 12 |
| | Account_Inquiry | 10 | 1 | 하 | 3 | |
| | Product_Inquiry | 9 | 1 | 하 | 3 | |
| | Credit_Inquiry | 6 | 1 | 하 | 3 | |
| 초기 Function Point 소계 | | | | | | 35 |
| 초기 Function Point 합계 | | 21 + 5 + 38 + 23 + 12 | | | | 99 |

단계 6 : 조정 인자 결정

■ VAF, Value Adjustment Factor

- 초기 FP 값에 대하여 대상 시스템의 특성을 반영하기 위함.

■ 14개 시스템 특성 항목

1. 데이터 통신(Data Communication)
2. 분산 데이터 처리(Distributed Data Processing)
3. 성능(Performance)
4. 비중 높게 사용되는 자원(Heavily Used Configuration)
5. 트랜잭션 처리 율(Transaction Rate)
6. 온라인 데이터 입력(Online Data Entry)
7. 단말 사용자 효과(End User Efficiency)
8. 온라인 갱신(Online Update)
9. 복잡한 처리(Complex Processing)
- :

단계 6 : 조정 인자 결정 (계속)

■ 14개 시스템 특성 항목 (계속)

10. 재사용성(Reusability)
11. 설치의 용이성(Installation Ease)
12. 운영의 용이성(Operational Ease)
13. 다중 사이트(Multiple Sites)
14. 변경의 지원성(Facilitate Change)

■ 항목 평가 스케일 : DI, Degree of Influence

| | | | |
|---|----------------|---|-----------|
| 0 | 표현되지 않거나 영향 없음 | 3 | 평균 정도의 영향 |
| 1 | 미비한 영향 | 4 | 중요한 영향도 |
| 2 | 보통 정도의 영향 | 5 | 매우 강한 영향도 |

6.1 데이터 통신

- 응용 시스템이 처리 프로세서와의 직접 통신 정도
- 데이터 통신의 특성 배점 기준

| 스케일 | 기 준 | 비 고 |
|-----|---|-----|
| 0 | 순수한 배치 처리나 단독의 PC 처리 | |
| 1 | 배치처리이며, 원격 자료 입력 또는 출력 처리가 존재 | |
| 2 | 배치처리이며, 원격 자료 입력과 출력 처리가 존재 | |
| 3 | 온라인 자료 수집 또는 배치나 질의에 대한 전처리 과정 존재 | |
| 4 | 다수의 전처리 과정이 존재하며, 원격 처리를 위한 1가지 통신 프로토콜만 사용 | |
| 5 | 다수의 전처리 과정이 존재하고, 원격 처리를 위한 다수의 통신 프로토콜 사용 | |

6.2 분산 데이터 처리

- 서브 시스템들간의 데이터 전송 정도
- 분산 데이터 처리의 특성 배점 기준

| 스케일 | 기 준 | 비 고 |
|-----|--|-----|
| 0 | 시스템 구성 요소들간의 데이터 또는 처리에 대한 공유를 지원하지 않음. | |
| 1 | 시스템의 다른 구성요소(스프레드 쉬트 등)에서 데이터를 처리할 수 있도록 데이터를 준비 | |
| 2 | 시스템의 다른 구성요소가 데이터를 처리할 수 있도록 전송 | |
| 3 | 분산처리 및 분산 데이터 전송이 단 방향 온라인에 의해 처리 | |
| 4 | 분산처리 및 분산 데이터 전송이 양 방향 온라인에 의해 처리 | |
| 5 | 데이터에 대한 처리 기능이 시스템의 적당한 구성요소 사에서 동적으로 수행됨. | |

6.3 성능

- 응용 시스템 개발에 영향을 줄 수 있는 응답시간과 처리 속도
- 성능에 대한 특성 배점 기준

| 스케일 | 기 준 | 비 고 |
|-----|--|-----|
| 0 | 사용자에 의해 제시된 성능관련 요구사항 없음 | |
| 1 | 제시된 성능 및 설계 요구사항이 검토되었으나 특별한 조치가 필요 없음. | |
| 2 | 피크 시간동안에 처리 능력이나 응답시간에 대한 성능이 요구되나, 이를 위해 처리기 관련 설계에는 영향이 없음 | |
| 3 | 모든 시간동안에 처리 능력이나 응답시간에 대한 성능이 요구되나, 이를 위해 처리기 관련 설계에는 영향이 없음 | |
| 4 | 사용자 성능 요구사항이 매우 엄격하여 설계 단계의 성능 분석 작업이 필수적 | |
| 5 | 사용자 성능 요구사항을 만족시키기 위하여 성능 분석 지원 도구를 활용해야 함. | |

6.4 비중 높게 사용되는 자원

- 응용 시스템 개발에 영향을 줄 수 있는 자원 제약의 정도
- 비중 높은 자원에 대한 특성 배점 기준

| 스케일 | 기 준 | 비 고 |
|-----|--------------------------------|-----|
| 0 | 어떠한 제약도 존재하지 않음 | |
| 1 | 운영상의 제약사항은 존재하나 특별히 고려될 사항은 없음 | |
| 2 | 다소의 보안사항과 시간 고려사항이 포함 | |
| 3 | 시스템 일부분에 대한 특별한 프로세서 요구사항이 존재 | |
| 4 | 제시하는 운영 제약사항이 특별한 제한 사항으로 고려됨 | |
| 5 | 분산된 시스템 구성 요소들간의 특별한 제한사항이 존재 | |

6.5 트랜잭션 처리율

- 시스템 개발에 영향을 줄 수 있는 트랜잭션 발생 및 처리율 정도
- 트랜잭션 처리율의 특성 배점 기준

| 스케일 | 기 준 | 비 고 |
|-----|---|-----|
| 0 | 트랜잭션 처리의 피크 기간이 없음. | |
| 1 | 월, 분기, 년 등의 트랜잭션 피크 처리 기간이 존재 | |
| 2 | 주 단위로 트랜잭션 피크 처리 기간이 존재 | |
| 3 | 매일 트랜잭션 피크 처리 기간이 존재 | |
| 4 | 사용자가 요구하는 높은 트랜잭션 처리율이 시스템의 설계에서 성능 분석이 필요할 정도로 높음 | |
| 5 | 높은 트랜잭션 처리율을 요구하고 있으며, 설계, 개발, 설치 단계에서 성능 분석을 위한 도구가 필요 | |

6.6 온라인 데이터 입력

- 온라인 트랜잭션에 의해 입력되어지는 데이터의 정도
- 온라인 데이터 입력의 특성 배점 기준

| 스케일 | 기 준 | 비 고 |
|-----|------------------------------|-----|
| 0 | 모든 트랜잭션이 배치 모드에 있음 | |
| 1 | 전체 트랜잭션의 1~7%가 양 방향 데이터 입력 | |
| 2 | 전체 트랜잭션의 8~15%가 양 방향 데이터 입력 | |
| 3 | 전체 트랜잭션의 16~23%가 양 방향 데이터 입력 | |
| 4 | 전체 트랜잭션의 24~30%가 양 방향 데이터 입력 | |
| 5 | 전체 트랜잭션의 30% 이상이 양 방향 데이터 입력 | |

6.7 단말 사용자 효과

- 사용자에게 제공할 기능의 용이성, 체계성, 효율성 정도
- 단말 사용자 효과의 시스템 특성 항목

- 네비게이션(Navigation) 지원(함수 키, 동적 생성 메뉴 등)
- 메뉴
- 온라인 도움말과 온라인 문서
- 커서의 자동 이동
- 스크롤링(Scrolling)
- 온라인을 통한 원격 출력
- 사전 정의된 함수 키
- 온라인 트랜잭션에 의해 시작되는 배치 작업
- 스크린 데이터에 대한 커서 선택
- 밑줄, 컬러 등을 이용한 강조 부분(Highlighting)에 대한 표시
- 온라인 트랜잭션에 의한 인쇄된 사용자 문서
- 마우스 인터페이스
- 팝업(Pop-Up) 윈도우
- 비즈니스 기능을 수행하기 위해 최소한의 화면
- 2개 국어의 지원
- 다국 언어의 지원

6.7 단말 사용자 효과 (계속)

■ 단말 사용자 효과의 특성 배점 기준

| 스케일 | 기 준 | 비 고 |
|-----|--|-----|
| 0 | 위 나열된 항목 중에 어느 것도 속하지 않음. | |
| 1 | 나열 항목 중 1~3개에 속함 | |
| 2 | 나열 항목 중 4~5개에 속함 | |
| 3 | 나열 항목 중 6개 이상에 속하나, 사용자 요구사항이 효과 측면에서 특별하지 않음 | |
| 4 | 나열 항목 중 6개 이상에 속하나, 사용자 요구사항이 효과 측면에서 설계에 고려되어야 함. | |
| 5 | 나열 항목 중 6개 이상에 속하나, 사용자 효과 측면을 지원하기 위해 특정 도구의 사용과 데모가 요구됨. | |

6.8 온라인 갱신

- 내부의 논리적 파일에 대한 온라인 갱신의 정도
- 온라인 갱신의 특성 배점 기준

| 스케일 | 기 준 | 비 고 |
|-----|---|-----|
| 0 | 없음 | |
| 1 | 1~3개의 제어 파일에 대한 온라인 갱신이 요구되며, 갱신의 양이 적고, 복구도 용이함. | |
| 2 | 4개 이상의 제어 파일에 대한 온라인 갱신이 요구되며, 갱신의 양이 적고, 복구도 용이함 | |
| 3 | 중요한 내부 논리 파일에 대한 온라인 갱신이 요구됨 | |
| 4 | 3 + 데이터 분실에 대한 대비책이 필수적으로 필요함 | |
| 5 | 3 + 복구시 높은 비용을 지불해야 하는 방대한 양의 데이터 존재 | |

6.9 복잡한 처리

- 응용 시스템의 처리 로직에 대한 복잡성 정도
- 복잡한 처리의 시스템 특성 항목

- 민감한 제어 처리 또는 응용 시스템 의존적인 보안 처리
- 비중 높은 논리적 처리
- 비중 높은 수학적 처리
- 많은 예외 처리
- 다종의 입력과 출력을 고려해야 하는 복잡한 처리

- 데이터 통신의 특성 배점 기준

| 스케일 | 기 준 | 비 고 |
|-----|--------------------|-----|
| 0 | 나열 항목에 속하지 않음 | |
| 1 | 나열 항목 중 적어도 1개에 속함 | |
| 2 | 나열 항목 중 적어도 2개에 속함 | |
| 3 | 나열 항목 중 적어도 3개에 속함 | |
| 4 | 나열 항목 중 적어도 4개에 속함 | |
| 5 | 나열 항목 중 적어도 5개에 속함 | |

6.10 재사용성

- 코드 및 처리 로직이 다른 응용시스템 개발에 사용될 수 있는 정도
- 재사용성 특성 배점 기준

| 스케일 | 기 준 | 비 고 |
|-----|----------------------------------|-----|
| 0 | 재사용 가능한 코드 없음 | |
| 1 | 응용 시스템 범주 내에서의 재사용 코드 | |
| 2 | 응용 시스템의 10%미만이 재사용되도록 요구됨 | |
| 3 | 응용 시스템의 10% 이상 재사용되도록 요구됨 | |
| 4 | 응용 시스템이 패키지화되고, 재사용이 용이하도록 문서화됨 | |
| 5 | 4 + 사용자 요청 기반 유지보수에 사용될 수 있도록 지원 | |

6.11 설치의 용이성

- 개발 시스템으로 사용자 운영환경으로 전환하는 과정의 복잡도
- 설치의 용이성에 대한 특성 배점 기준

| 스케일 | 기 준 | 비 고 |
|-----|--|-----|
| 0 | 특별한 고려사항이나 셋업이 필요 없음 | |
| 1 | 사용자가 요구하는 특별한 고려사항은 없으나 특별한 셋업 필요 | |
| 2 | 매우 중요하지는 않지만 사용자에게 의해 요구되는 변환과 설치 요구사항이 있음 | |
| 3 | 사용자가 요구하는 변환과 설치 요구사항이 비중있게 고려됨 | |
| 4 | 2 + 자동화된 변환 및 설치 지원 도구가 필요 | |
| 5 | 3 + 자동화된 변환 및 설치 지원 도구가 필요 | |

6.12 운영의 용이성

- 응용 시스템이 초기 구동, 백업, 결함 복구 등의 운영 측면 지원 정도
- 운영의 용이성에 대한 시스템 특성 항목

- 효과적인 시동, 백업, 복구 절차가 필요하며, 사용자 간섭이 요구됨
- 효과적인 시동, 백업, 복구 절차가 필요하나, 사용자 간섭이 요구되지 않음
- 응용 시스템이 테이프와 같은 장치 요구를 최소로 함
- 응용 시스템이 용지 처리와 같은 요구를 최소로 함

- 데이터 통신의 특성 배점 기준

| 스케일 | 기 준 | 비 고 |
|-----|--|-----|
| 0 | 일반적인 백업절차를 제외하면 사용자가 요구하는 사항 없음 | |
| 1 | 나열 항목 중 1개가 해당 | |
| 2 | 나열 항목 중 2개가 해당 | |
| 3 | 나열 항목 중 3개가 해당 | |
| 4 | 나열 항목 중 4개가 해당 | |
| 5 | 운영자의 간섭 없이 자동 백업이나, 오류 자동 복구 같은 작업이 요구됨. | |

6.13 다중 사이트

- 개발 시스템이 얼마나 많은 위치와 사용자 조직에서 사용할 것인가
- 다중 사이트의 특성 배점 기준

| 스케일 | 기 준 | 비 고 |
|-----|--|-----|
| 0 | 단지 한 사이트에 설치되는 것을 요구함. | |
| 1 | 설계 과정에서 다수의 사이트에 대한 필요성이 요구되나, 동일한 H/W와 S/W 환경이 요구됨 | |
| 2 | 설계 과정에서 다수의 사이트에 대한 필요성이 요구되며, 유사한 H/W와 S/W 환경이 요구됨 | |
| 3 | 설계 과정에서 다수의 사이트에 대한 필요성이 요구되나, 서로 다른 H/W와 S/W 환경이 요구됨 | |
| 4 | 다수의 사이트에 응용 시스템을 설치하기 위한 문서화 및 지원 계획이 요구되며, 동일 또는 유사한 H/W와 S/W 환경이 요구됨 | |
| 5 | 다수의 사이트에 응용 시스템을 설치하기 위한 문서화 및 지원 계획이 요구되며, 서로 다른 H/W와 S/W 환경이 요구됨 | |

6.14 변경의 지원성

- 처리 로직이나 데이터 구조 측면에서의 변경 수용의 용이성
- 변경의 지원성에 대한 특성 배점 기준

| 스케일 | 기 준 | 비 고 |
|-------|---|-----|
| 0 | 사용자가 요구하는 특별한 사항이 존재하지 않음 | |
| 1 - 5 | <ul style="list-style-type: none"> - 단순한 요청사항을 처리할 수 있는 융통성 있는 질의 및 보고 기능 제공 : 배점 1로 계산 - 보통의 요청 사항을 처리할 수 있는 융통성 있는 질의 및 보고 기능 제공 : 배점 2로 계산 - 복잡한 요청 사항을 처리할 수 있는 융통성 있는 질의 및 보고 기능 제공 : 배점 3으로 계산 - 비즈니스 제어 데이터가 사용자가 사용하는 온라인 상호 작용 테이블에 유지되며, 변경이 다음 단계의 업무에 다소 영향을 줌. : 배점 1로 계산 - 비즈니스 제어 데이터가 사용자가 사용하는 온라인 상호 작용 테이블에 유지되며, 변경이 즉시 처리됨 : 배점 2로 계산 | |

단계 6 : 조정 인자 결정 (계속)

- **TDI, Total Degree of Influence : 14 항목 배점의 합계**
- **VAF 산출**
 - $VAF = (TDI \times 0.01) + 0.65$
- **일반적인 시스템에서의 TDI 산출치 예**
 - 단순 배치 시스템 : > 15
 - 전처리 기능을 갖는 배치 시스템 : 15 ~ 30
 - 상호 작용이 많은 응용 시스템 : 30 ~ 45
 - 실시간 통신 및 프로세스 제어 시스템 : 30 ~ 60
- **일반적인 시스템에서의 VAF 산출치 예**
 - 전처리 기능을 갖는 배치 시스템 : 0.8 ~ 0.85
 - 실시간 및 통신 시스템 : 0.95 ~ 1.25

양식 9. FP 조정 인자 값 산출

| 번호 | 시스템 특성 | 평가 배점 | 비고 |
|------------|---------------|-------|----|
| 1 | 데이터 통신 | 4 | |
| 2 | 분산 데이터 처리 | 0 | |
| 3 | 성능 | 0 | |
| 4 | 비중 높게 사용되는 자원 | 0 | |
| 5 | 트랜잭션 처리 율 | 4 | |
| 6 | 온라인 데이터 입력 | 5 | |
| 7 | 단말 사용자 효과 | 3 | |
| 8 | 온라인 갱신 | 3 | |
| 9 | 복잡한 처리 | 1 | |
| 10 | 재사용성 | 1 | |
| 11 | 설치의 용이성 | 1 | |
| 12 | 운영의 용이성 | 4 | |
| 13 | 다중 사이트 | 1 | |
| 14 | 변경의 지원성 | 2 | |
| TDI | | 29 | |

- $VAF = (29 \times 0.01) + 0.65 = 0.94$

단계 7 : 최종 FP 산출

- AFP, Adjusted Function Point
- $AFP = UAF \times VAF$

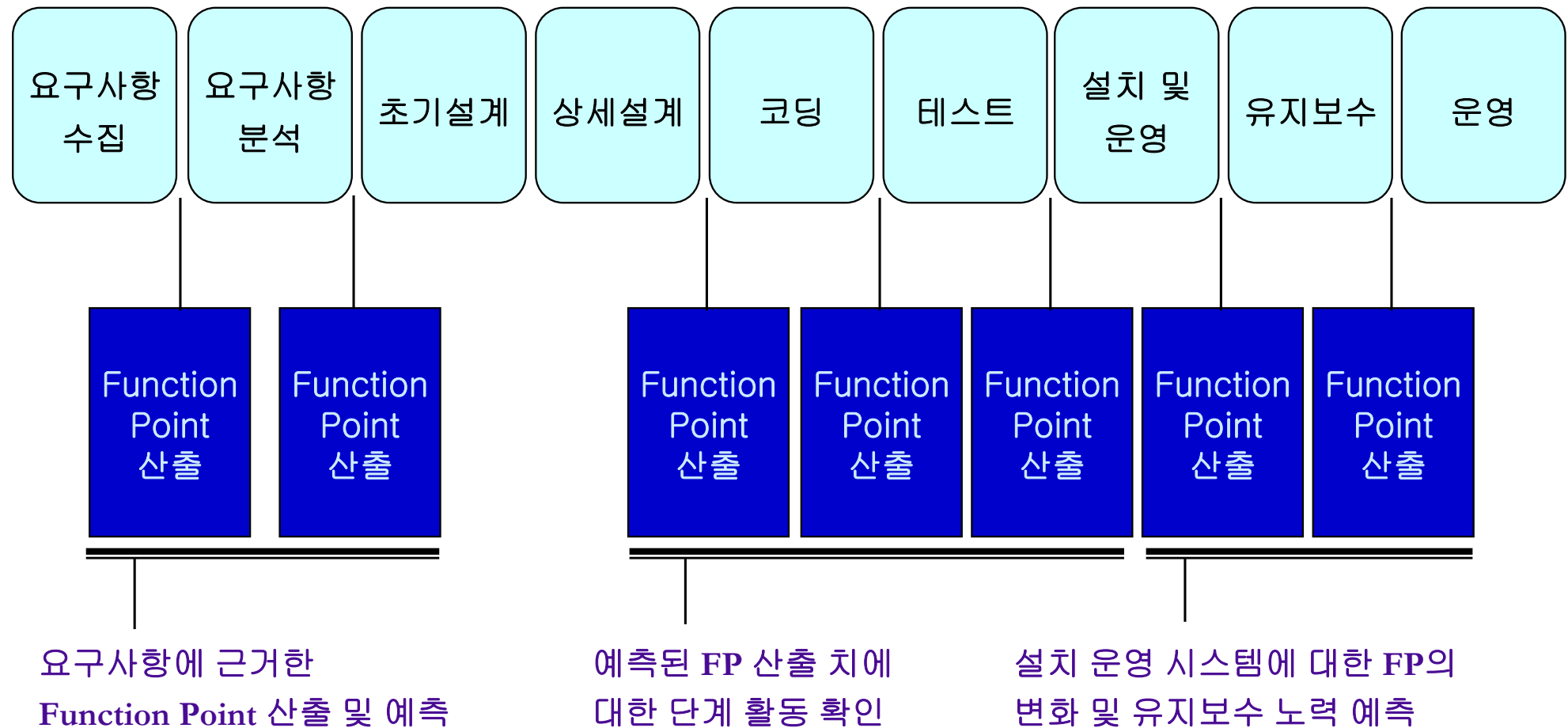
| UAF | VAF | AFP |
|-----|------|------|
| 99 | 0.94 | 93.1 |



프로젝트 타입별 FP 산출

프로젝트 단계별 FP 산출 시점

■ 소프트웨어 개발 생명주기 단계



응용 시스템 개발 프로젝트

■ Development Function Point,

$$DFP = (UFP \times CFP) \times VAF$$

- CFP : Function Point for data Conversion

■ Enhancement Function Point,

$$EFP = [(ADD + CHGA + CFP) \times VAFA] + (DEL \times VAFB)$$

- ADD : 신규 추가되는 부분에 대한 UFP
- CHGA : 변경되는 기능에 대한 UFP
- VAFA : 개선 프로젝트 수행 후, 시스템에 대한 조정인자 값
- DEL : 삭제되는 기능에 대한 UFP
- VAFB : 개선 프로젝트 수행전의 시스템 조정인자 값

운영 시스템의 Function Point 산출

■ Application Function Point,

$$AFP = ADD \times VAF$$

- ADD : 개발 프로젝트에 의해 설치된 기능의 UFP

■ Application Function Point after Enhancement,

$$AFP = [(UFPB + ADD + CHGA) - (CHGB + DEL)] \times VAFA$$

- UFPB : 개선 프로젝트 수행 전의 UFP
- ADD : 개선 프로젝트에 의해 추가된 기능의 UFP
- CHGA : 개선 프로젝트에 의해 변경된 부분의 UFP
- CHGB : 개선/변경된 기능의 본래 UFP
- DEL : 개선 프로젝트에 의해 삭제된 기능의 UFP
- VAFA : 개선 프로젝트 수행 후, 시스템 조정인자 값

Function Point 척도의
활용 및 이득

Function Point 관련 척도

■ 생산성 척도

- 주어진 일정상에서 한 FP를 개발하기 위한 소요 시간
- 신규개발 생산성 VS. 기존시스템 개선생산성

$$\frac{\text{프로젝트 총 소요 시간}}{\text{산출된 FP 값}}$$

■ 품질 척도

- 요구사항 변경에 따른 FP 변경 값의 측정
 - 요구사항 변경율 및 프로젝트 변경율
- 수명주기 개발 단계에 대한 노력 비교

$$\frac{\text{수명주기 단계별 결함수}}{\text{산출된 FP 값}}$$

Function Point 관련 척도 (계속)

■ 비용 척도

- 한 FP를 개발하기 위한 소요 비용
- 서로 다른 개발 단계에서의 FP당 비용 측정
- 대응하는 상용 시스템에 대한 비용 비교

- 응용 시스템의 오류 수정을 위해 소요되는 비용 측정
- 신규 설치 시스템에 대한 6개월간의 소요 비용 추정

$$\frac{\text{총 비용}}{\text{산출된 FP 값}}$$

$$\frac{\text{총 수정시간} \times \text{시간당 비용}}{\text{배포된 최종 FP 값}}$$

Function Point 관련 척도 (계속)

■ 유지보수 척도

- 핵심 S/W 또는 관리 대상 S/W의 유지보수 모니터링
- 일정주기 동안의 시스템 기능에 대한 변화/성장율
- 시스템 설치 2 ~ 3개월간의 시스템 안정도 측정

$$\frac{\text{유지보수 비용}}{\text{산출된 FP 값}}$$

$$\frac{\text{현재 FP 값}}{\text{본래 초기 FP 값}}$$

$$\frac{\text{변경 횟수}}{\text{응용 시스템 FP 값}}$$

Function Point 적용 이득

- 소프트웨어 개발 프로젝트 계획 측면
 - 요구사항에 대한 정확한 이해 증대
 - 소요 기간 및 소요 자원 산정의 정확성 향상
 - 개발 비용 산정의 정확성 향상

- 소프트웨어 개발 프로젝트 관리 측면
 - 소프트웨어 개발 생산성 관리
 - 소프트웨어 개발 베이스라인별 품질 측정

- 소프트웨어 개발 조직 관리 측면
 - 운영 소프트웨어에 대한 수명주기 관리
 - 운영 유지에 대한 계획 및 관리

프로그래밍 언어 LOC/FP

■ 1 FP당 구현언어 라인 수 비교

| 구현 언어 | | 라인 수 | 구현 언어 | 라인 수 |
|--------------|-------|------|---------------|------|
| Assembly | Basic | 320 | FORTRAN | 107 |
| | Macro | 213 | | |
| BASIC | | 107 | HTML 3.0 | 15 |
| Visual Basic | | 29 | LISP | 64 |
| C | | 128 | JAVA | 53 |
| C++ | | 53 | PL/I | 80 |
| COBOL | | 107 | SQL | 13 |
| DELPHI | | 29 | Power Builder | 16 |

* Source : www.theadvisors.com/langcomparison.htm



결론 및 요약

결론

■ Function Point

- 개발 소프트웨어 시스템에 대한 기능 사이즈 측정 방법
- 프로젝트 관리 요소에 대한 예측 방법 지원
 - 프로젝트 소요 자원, 일정, 생산성
- FP 이외의 다양한 척도 활용 : 프로젝트 전체 평가 가능

■ Function Point 산출 기법

- 정규법
 - ILF, EIF, EI, EO, 및 EQ에 대한 DET, RET, FTR를 식별
 - 식별된 복잡도에 대하여 FP 산출
- 간이법
 - ILF, EIF, C/R/U/D/O에 대한 복잡도 산출
 - 식별된 복잡도에 대한 FP 산출

결론 (Cont.)

- **Function Point 산출치의 신뢰성 향상 방안**
 - Wide-Band Delphi 방법에 의한 전문가 의견 반영
 - 간이법에서의 기능 수준 가중치 세분화

- **Function Point 산출치의 활용**
 - Function Point에 의한 산출치 해석
 - 히스토리 데이터에 대한 축적 및 활용
 - 프로젝트 유형에 따른 FP 산출 방식의 점진적 적용
 - 척도 저장소를 통한 FP 관리 및 활용

- **대표적인 FP 산출 지원 도구**
 - COSMOS : Software Cost Modeling System, U.S.A Defense
 - Function Point WORKBENCH, Melbourne, Australia

첨부 : 약어

- **ADD** : Function Point for Newly Addition
- **AFP** : Adjusted Function Point

- **CFP** : Function Point for data Conversion
- **CHGA** : Function Point After Change
- **CHGB** : Function Point Before Change
- **COSMOS** : software COSt Modeling system

- **DET** : Data Element Type
- **DEL** : Function Point for Deletion
- **DI** : Degree of Influence

- **EI** : External Input
- **EIF** : External Interface File
- **EO** : External Output
- **EQ** : External inQuery

- **FP** : Function Point
- **FTR** : File Type Referenced

