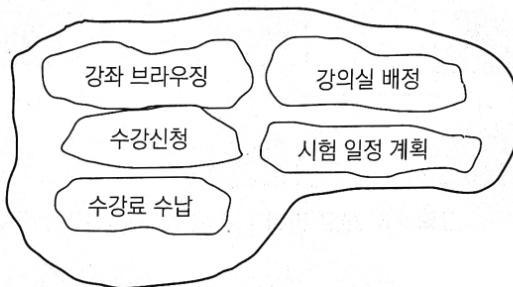
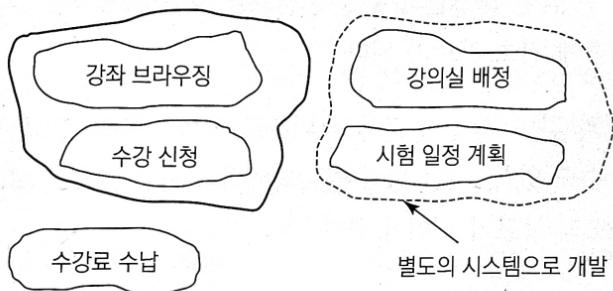


문이다.

범위를 정하는 방법은 시스템이 해결할 것으로 예상되는 문제들을 모두 리스트로 적는 것이다. 범위를 축소 하려면 문제 리스트에서 몇 가지를 삭제하여 별도의 프로젝트로 남겨둔다. [그림 3.5]에 있는 것이 범위 축소 과정을 나타낸 것이다.



(a) 광범위한 문제 범위



(b) 축소된 범위

그림 3.5 수강 신청 시스템 프로젝트 범위

### 3.2.3 WBS

프로젝트의 목표를 스케줄 관점으로 이해하기 위하여 WBS(Work Breakdown Structure) 분석이 필요하다. WBS는 개발 팀이 프로젝트 목표를 달성하고 결과물을 산출하기 위하여 수행하여야 할 작업을 계층적으로 분할한 것이다. 이러한 소작업들에 대한 소요 일정이 예측되어야 전체 프로젝트의 일정을 계획할 수 있다.

세부적으로 나누어진 WBS는 [그림 3.6]과 같이 나타낼 수 있다. ‘재고 관리 시스템’ 개발 프로젝트의 주요 작업이 루트 아래 표시되어 있다. 트리의 모든 노드는 자식 노드에 표시된 소작업으로 나누어진다. WBS를 작성하는 목적은 프로젝트 진행에서 일어나는 모든 작업을 찾아내기 위한 것이다. 각 작업은 필요에 따라 더 자세한 소작업으로 나눌

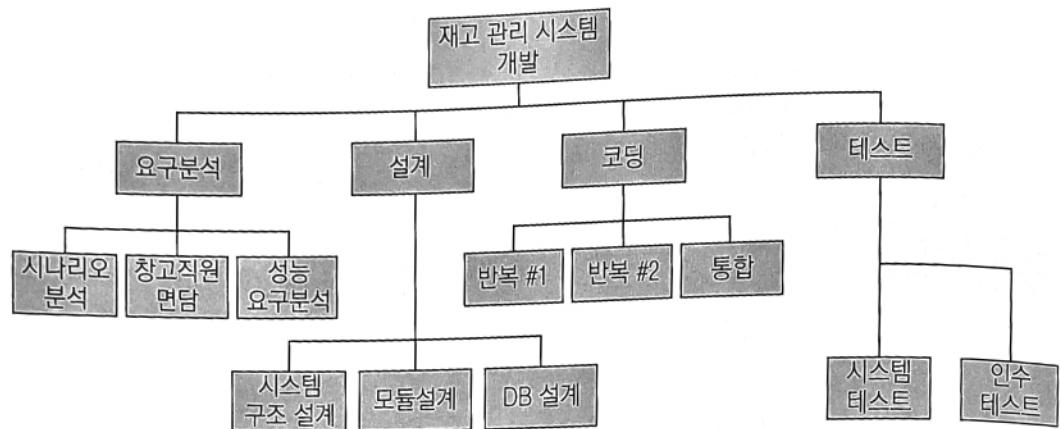


그림 3.6 재고 관리 시스템 개발을 위한 WBS

수 있다. 예를 들면, ‘설계’ 작업은 시스템 구조 설계, 모듈 설계, DB 설계로 이루어진다. WBS에서 최하위 단말 노드들의 수행에 필요한 자원의 예측은 프로젝트 전체 수행에 필요한 자원 예측에 기초 자료가 된다.

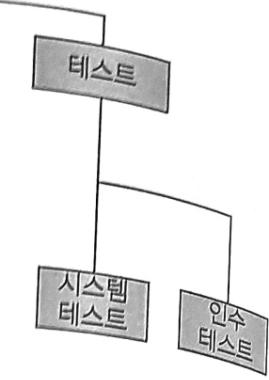
WBS는 [그림 3.6]과 같이 작업 프로세스 위주로 작성할 수도 있고 작업 단위를 산출물 위주로 작성할 수도 있다. 어떤 형식이든 최하위의 단위 작업은 프로젝트 단위 조직이 책임져야 할 규모의 목표가 되어야 한다.

WBS는 프로젝트에서 수행할 작업을 정의하는 데 효과적이며 목표와 관련된 작업을 잘 보여준다. 또한, 작업과 비용을 정의하고 담당을 정하고 모니터링하는 데 유용하다. 또한 작업에 대하여 책임을 진 조직의 부서를 파악하는데 도움이 된다.

WBS는 프로젝트 스케줄링 작업의 입력이 된다. WBS를 통하여 프로젝트가 어떤 작업으로 이루어지는지를 알아낼 수 있고 스케줄링은 이들 작업을 어떤 순서로 할 것인가를 정하는 작업이다. WBS의 각 노드에 표시된 항목은 프로젝트에서 수행해야 할 작업이다. 스케줄링은 빠른 기간 내에 프로젝트를 완성할 수 있도록 이러한 작업의 순서를 최적화하는 일이다.

### 3.2.4 스케줄링

프로젝트의 목표가 설정된 후 하여야 할 일은 WBS를 기초로 하여 일정을 정의하는 것이다. 일정을 파악하여야 프로젝트에 투입될 인력의 규모와 비용을 추정할 수 있기 때문이다.



DB 설계로 이루어진  
프로젝트 전체 수행

고 작업 단위를 산출  
· 프로젝트 단위 조직

표와 관련된 작업을  
링하는 데 유용하다.  
된다.

로젝트가 어떤 작업  
면 순서로 할 것인가  
서 수행해야 할 작업  
러한 작업의 순서를

일정을 정의하는 것  
추정할 수 있기 때

스케줄링 결과는 간트 차트로 표현되는데 다음과 같은 순서의 작업이 필요하다.

- 1) 작업 사이의 의존 관계 파악
- 2) CPM 방법을 이용한 여유 시간 계산
- 3) 소요 자원의 할당

#### ■ 작업 의존 관계

작업을 수행하는 순서는 논리적으로 정해져 있다. 예를 들어 집을 지을 때 기초 공사를 하기 전에 지붕을 올릴 수는 없다.

작업 사이의 의존관계는 강한 관계와 약한 관계가 있다. 코딩한 후에야 테스팅 할 수 있다든지 계약이 끝나야 라이브러리를 사용하여 코딩을 시작할 수 있는 것이 강한 관계이다. 강한 의존관계를 제거하기 위해서는 재작업 비용이 든다.

약한 의존관계는 순서에 따라 다른 계획이 가능하다. 예를 들어 어떤 모듈을 먼저 개발할 것인지 순서를 정하는 것은 약한 의존관계이다. 약한 의존관계는 사용할 수 있는 자원과 수준에 따라 다른 순서로 진행할 수 있다.

작업 의존 관계와 소요 시간을 표로 나타내면 [표 3.1]과 같이 나타낼 수 있다. 선행 작업이란 어떤 작업을 시작하기 위하여 미리 완료되어야 하는 작업을 말한다. 예를 들면 C 작업의 선행 작업이 A로 표시된 것은 C 작업이 시작되기 전에 A 작업이 끝나야 한다

표 3.1 작업 의존 관계

작업	선행 작업	소요 기간(일)
A	-	8
B	-	15
C	A	15
D	-	10
E	B, D	10
F	A, B	5
G	A	20
H	D	25
I	C, F	15
J	G, E	15
K	I	7
L	K	10

는 것을 의미한다. 즉, C가 소프트웨어 구조 설계 작업이라 한다면 선행 작업인 요구 분석 A 작업이 반드시 끝나 있어야 한다. 이러한 관계는 WBS 자체에는 나타나지 않는다.

### ■ CPM 네트워크

CPM(Critical Path Method) 네트워크는 노드와 간선으로 구성된 네트워크이다. CPM 네트워크는 여러 가지 변형이 있다. 노드에는 작업을 표시하고 간선은 작업 사이의 선후 의존 관계를 나타낸다. 간선을 나타내는 화살표의 머리에 있는 작업은 화살표의 꼬리에 있는 작업이 끝날 때까지는 시작될 수 없다.

WBS에서와 같이 CPM 네트워크의 박스에는 작업의 시작일과 완성일을 표시한다. 이 것으로 각 작업에 대한 가장 이른 시작일(Earliest Starting Time: ES)을 구할 수 있다. 일부 노드는 이정표(milestones)로 지정할 수 있다. 이정표는 프로젝트의 중요한 중간 결과를 완성하였다는 표시이다. 만일 이정표의 일을 완성시키지 못하였다면 일정을 수정하여야 한다는 것을 의미한다.

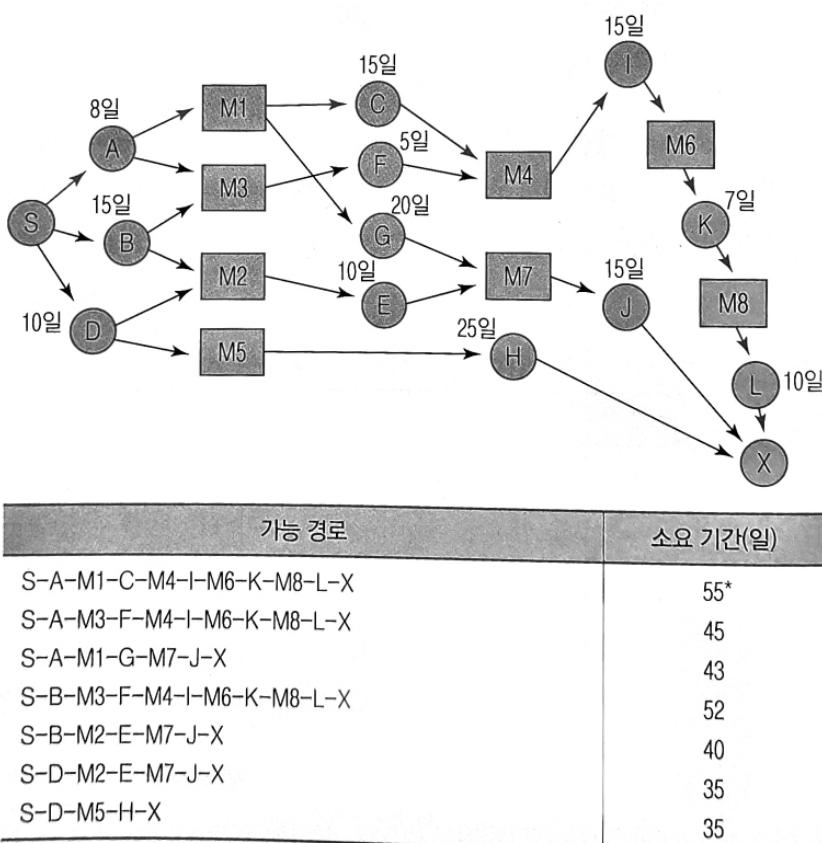


그림 3.7 CPM 네트워크와 임계 경로

CPM 네트워크는 프로젝트 완성에 필요한 작업을 나열하고 작업에 필요한 소요 기간을 예측하는 데 사용한다. 이를 위하여 각 작업의 선후 관계를 결정하여야 한다. CPM 네트워크는 어떤 작업이 필요하고 각각이 얼마나 걸리며 각 작업의 선후관계를 한눈에 볼 수 있도록 나타내어 전체 프로젝트의 최소 소요 기간을 구하는 데 사용한다.

### ■ 임계 경로와 여유 시간 계산

[그림 3.7]의 예에서 CPM 네트워크를 분석하면 S-A-M1-C-M4-I-M6-K-M8-L-X 작업으로 이루어진 경로가 소요 기간(55일)이 가장 긴 경로, 즉 임계 경로(critical path)가 된다. 임계 경로는 이 경로 상에 있는 어떤 작업이라도 늦추어지면 전체 프로젝트가 지연된다는 것을 의미한다. 관리자는 다른 작업보다 이런 작업들에 보다 관심을 두고 점검하여야 한다.

다음은 CPM 네트워크를 이용하여 각 작업의 여유 시간(slack time)을 구한다. 여유 시간은 다음의 식에 의하여 구할 수 있다.

$$TS = TL(\text{Latest Start Time}) - TE(\text{Earliest Start Time})$$

TE는 그 작업을 시작할 수 있는 최단 시간이다. CPM 네트워크를 이용하여 각 작업이 최대한 빠르게 시작할 수 있는 날짜, TE를 계산할 수 있다.

예를 들어 작업 C의 최대로 빠른 착수일, TE는 8일이 지난 다음 날이다. 여기서 15일 지난 후, 즉 23일이 경과하여야 작업 C를 끝낼 수 있다. 한편 작업 C를 최대로 늦추어 시작할 수 있는 시간, TL은 최대경로(55일)에서 C(15일), I(15일), K(7일), L(10일)의 총 경과 기간을 뺀 8일이다. 최대로 빠른 착수일(TE)과 최대로 늦추어 시작할 수 있는 날(TL)이 같으므로 여유 기간이 없다. 따라서 C 작업은 임계 경로에 있음을 알 수 있다.

다른 작업 E의 최대 빠른 착수일(TS)과 늦은 착수일(TL)을 구해보자. 작업 E는 B가 끝나는 15일 이후에야 착수할 수 있다. 늦은 착수일(TL)은 55일에서 E(10일), J(15일)의 경과 기간을 뺀 30일이며, 즉 15일의 여유 기간(slack time)이 있다. E 작업은 임계 경로에 있지 않기 때문이다.

## ■ 자원 할당과 간트 차트

각 작업에 대한 여유 시간을 구한 후 작업별로 시작과 종료 기간을 수평 막대로 표현한 것이 [그림 3.8]과 같은 간트 차트이다. 프로젝트 관리 도구를 이용하면 간트 차트를 자동으로 그릴 수 있다. 여기서 작업 2는 3주의 여유 시간이 있다는 것을 알 수 있다.

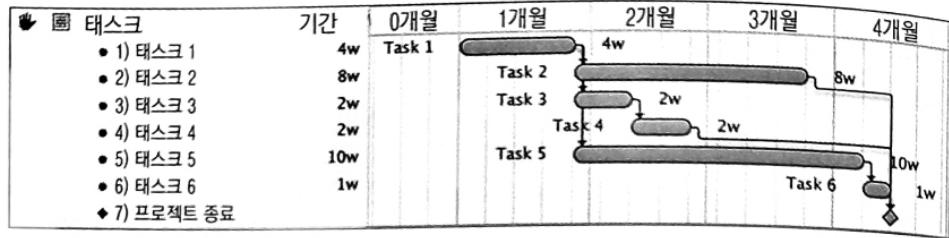


그림 3.8 간트 차트

프로젝트 진행을 위하여 일정 계획에는 필요한 자원을 할당한다. 소프트웨어 개발에 필요한 자원은 크게 세 가지로 나눌 수 있다.

- 인력 - 주어진 작업을 수행할 인원과 투입률
- 장비 - 주어진 작업을 수행할 때 필요한 도구나 하드웨어 및 소프트웨어
- 재료 - 주어진 작업을 수행할 때 필요한 소모품이나 자료

인력은 간트 차트에 표시된 작업 단위로 할당한다. 예를 들어 [그림 3.9]의 간트 차트에서는 작업 1에 개발자 홍길동을 50% 투입하고 작업 2에는 개발자 김동국을 100% 투입하였다. 소요 자원은 인력이 작업에 투입된 시간으로 나타낸다. 간트 차트의 각 열에는 투입된 자원의 총량이 표시된다.

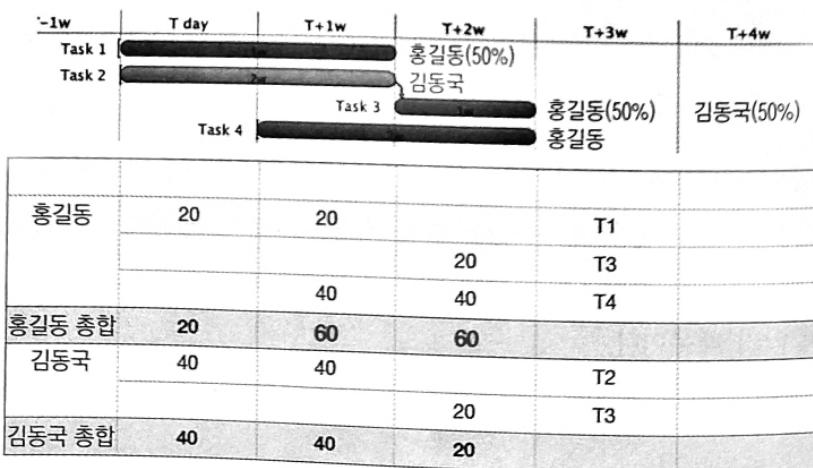


그림 3.9 소요 자원의 할당

### 애자일 프로세스의 스케줄링

CPM 네트워크와 간트 차트를 사용하는 전통적인 프로세스의 일정 계획과는 달리 애자일 프로세스는 스토리 카드와 같은 도구를 사용한다. 스토리 카드에는 사용자 스토리, 비즈니스 우선순위, 스토리 점수, 연관된 스토리 등이 적혀있다. 스토리 카드는 단어 암기 카드나 포스트잇 같은 것을 이용하면 우선순위나 연관성 등을 기준으로 쉽게 정렬할 수 있다.

애자일 프로세스의 일정 계획은 [그림 3.10]과 같이 카드나 포스트잇으로 빠르고 쉽게 만들 수 있다. 사용자 스토리를 적은 카드를 반복 단계별로 정리하여 수직으로 나열한다. 카드에 적힌 스토리 점수는 유스케이스를 개발하기 위하여 소요되는 노력을 의미한다. 스케줄링을 위하여 애자일 계획에서도 생산성 데이터가 필요하다. 이 팀은 스토리 점수 3점을 개발하려면 1주가 소요되었다.



그림 3.10 애자일 프로세스의 스케줄링

애자일 계획은 다음과 같은 장점이 있다.

- 높은 우선순위를 가진 유스케이스가 조기에 개발되어 설치된다는 확신을 줄 수 있다.
- 유스케이스 사이에 선행 관계를 지킬 수 있다. [그림 3.10]에서 UC3은 반복 주기 #2에 할당된 이유는 UC1과 UC9가 선행되어야 하기 때문이다.
- 각 열의 점수의 합은 개발 팀의 작업 속도를 초과하지 않아야 한다. 작업 속도는 생산성 통곗값으로 팀이 반복 주기를 완성하는 데 드는 평균 노력이다.

### 3.3 비용 예측 기법

프로젝트 계획 작업에서는 소프트웨어를 개발하기 위하여 드는 비용이 얼마이며 어느 정도의 기간이 걸리는지를 예측할 필요가 있다. 이러한 예측은 개발이 시작되기 전에