

Markup Develop

Process & Methodology

Process

Specification

Scheduling

**Coding
convention**

Quality

1. 개발명세 파악

- 1.1. 마크업 스펙 질의서
 - 1.1.1. 질의서 작성 가이드
- 1.2. 마크업 스펙 확정서

2. 일정산출

- 2.1. 목적
- 2.2. SO
- 2.3. 기획서 분석
- 2.4. 일정산출
- 2.5. 일정표 작성
- 2.6. 배포

3. 마크업 개발

- 3.1. 마크업 코드의 버전 관리
- 3.2. 마크업 코딩 컨벤션
 - 3.2.1. NULI
- 3.3. UIO
- 3.4. Pair Programming
- 3.5. HTS
 - 3.5.1. 개념
 - 3.5.2. 프로세스
 - 3.5.3. 모듈라이브러리
- 3.6. 마크업 메뉴얼
 - 3.6.1. 활용
- 3.7. 롤플레이닝
- 3.8. 릴리즈노트

4. 품질인증

- 4.1. 수치화
 - 4.1.1. 의미
 - 4.1.2. 적정지표
- 4.2. QP
 - 4.2.1. 개념
- 4.3. 체크리스트
- 4.4. 생산성 효율

1. 개발명세 파악

• 마크업 스펙 질의서

- 유관부서와의 **사전 협의**가 목적이며, 주요 내용은 아래와 같음

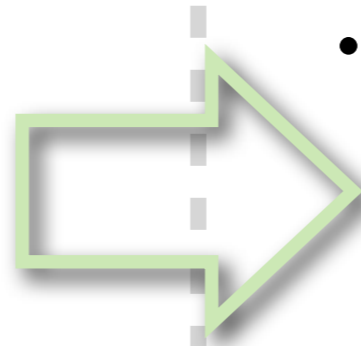
- W3C Validation 수준
- DTD 설정
- Character Set 설정
- 크로스브라우징의 범위

• 문서화의 그 시작!

- 제안의 필요성
 - 상호 협의된 기준의 부재
 - 불합리한 롤백의 최소화
 - HTML/CSS 이슈 부각 및 표면화
- 무엇을 어떻게 제안할 것인가?
 - Nothing to lose!
 - 유연한 리소스 환경 조성
 - 디자인 개발간 의존도 최소화 (구조와 표현의 분리)
 - 우선순위 파악 / 조정
 - 제안 그리고 대안
 - 제너럴리스트는 필수 조건?

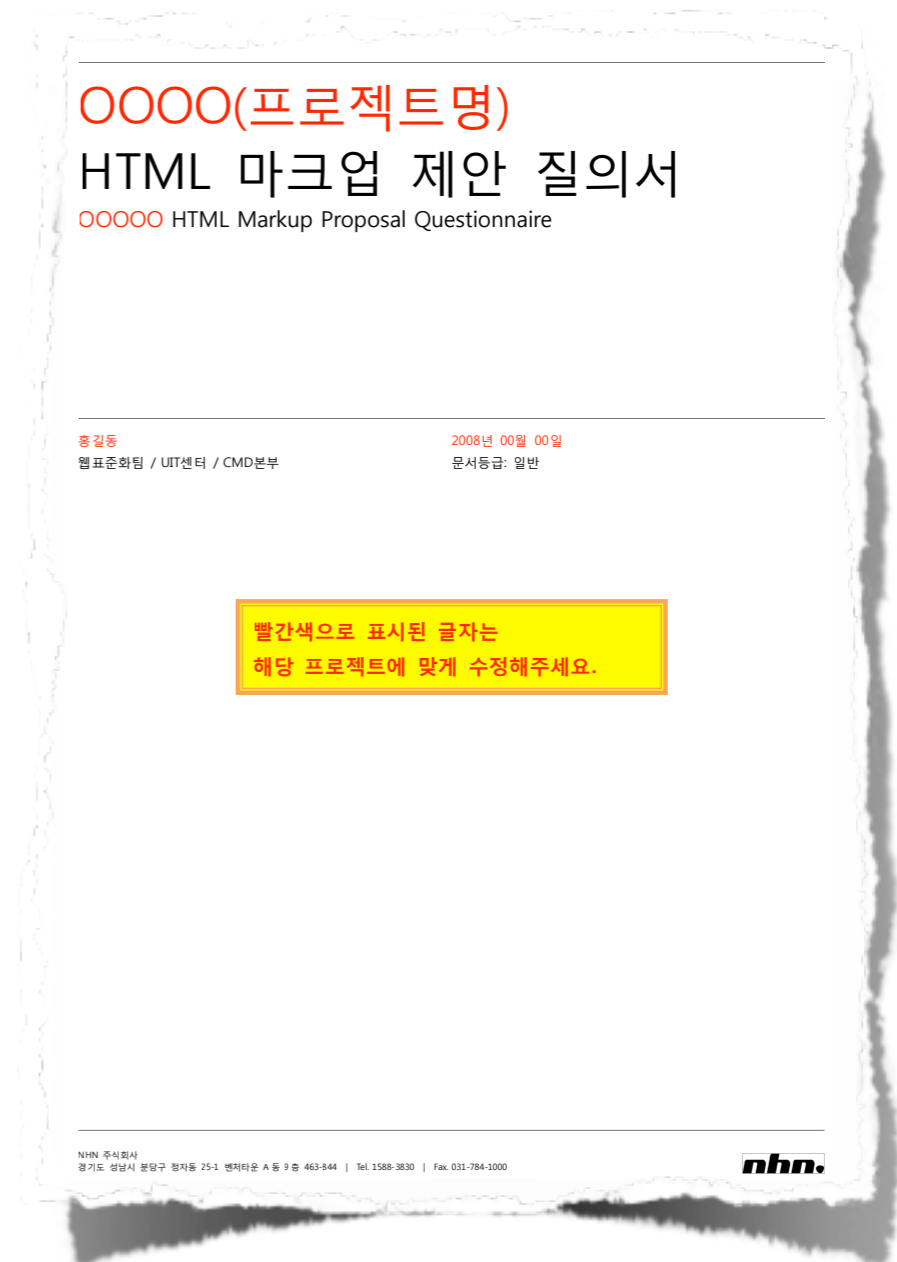
• Design Architecture

- 기획, 디자인, 개발을 아우르는 프로세스 전반의 시각 제공
- 문제의 소지들을 사전에 파악하여, 리스크를 최소화 함



• 마크업 스펙 확정서

- 질의서를 통해 확정된 내용으로 마크업 개발을 시작함
 - 각 담당 부서의 답변을 토대로 마크업 스펙설정의 히스토리를 관리함
 - 파일은 유관 부서원들이 언제나 열람 가능한 형태로 제공



2. 일정산출

- 일정산출의 목적

- 객관적 지표의 필요성

- 체계적이지 못한 스케줄링의 경우 일정의 작업자를 맞추는 경우가 대부분

- Why?

- 변하는 프로젝트 일정.
- 리소스 투입률의 변동
- 스펙아웃, 스펙인
- 작업자의 숙련도

- 작업자들간의 모호하지 않은 목표의식

- 체계적인 공식에 의해 **동의**된 일정
- 무엇을 해야 할지 구체적으로 알게 됨

- 강제 작용(Forcing function)

- 세밀하게 짜여진 일정은 그것을 공식화 하고 증폭하는 심리적인 효과가 발생하여, 작업자로 하여금 책임감을 높게 함

- 업무진행 상황의 추적

- 세분화된 일정단위는 할 일을 이해하기 쉽게 만들며, 그것을 추적하기 쉽게 하여, 결국 일정관리를 도모함
 - 어떤 부분이 가장 비용이 많이 들고,
 - 우선순위는 어떠한지, 무엇을 먼저 해야 하는지 알게 됨

HTML개발 일정 산출의 수치화

| 산출방식의 개요

1. 디자인페이지를 기준으로 구성요소별로 나눈다.
2. 나누어진 각 구성요소에 일정단위[so]를 적용한다.
3. 각 요소에 일정단위가 적용되면 기준단위를 정하고 소요시간을 산정한다.
4. 다른 요소별 단위값을 기준단위와의 비율을 산정하여 각 요소별 소요시간을 산정한다.
5. 요소별 소요시간의 합을 구하여 전체 소요시간을 산정한다

HTML개발 일정 산출의 수치화

산출방식의 예시

구성요소별로 나눈후 제작단계는
아래 내용을 기준으로 한다.

1. Layout 구조설계
2. Design Style CSS 설계
3. Top영역 (GNB + Menu)
4. 좌 우측 메뉴
5. 콘텐츠 영역
6. 크로스브라우징
7. 최적화 검수

작업순서



HTML개발 일정 산출의 수치화

산출방식의 예시

일정산출 비교값

1_{so} 2_{so} 3_{so} 5_{so} 8_{so} 13_{so} 20_{so}

피보나치 수열 적용
SO = Story + Object

$$5so = 2h \rightarrow 1so = 0.4h$$

$$\text{소요so} \cdot h/so = \text{요소별 소요시간}$$

h	so	
(1.2h)	(3so)	Layout 구조설계
(2h)	(5so)	<u>Design Style CSS 설계</u>
(3.2h)	(8so)	Top영역 (GNB + Menu)
(3.2h)	(8so)	좌 우측 메뉴
(5.2h)	(13so)	컨텐츠 영역
(0.8h)	(2so)	크로스브라우징
(0.4h)	(1so)	최적화 검수
15h	40so	

NULLI

NHN UI Library
<http://html.nhndesign.com>

마크업 기술 전반에 걸친
내용들을 매뉴얼화

산발적인 코딩 방식이
통일 되도록 유도

3. 코딩컨벤션

1. 개요	8
1.1 코딩 컨벤션 필요성	9
1.2 코딩 컨벤션 요소	10
1.3 코딩 컨벤션 용어	11
1.3.1 문법 용어	11
2. 기초 지식	12
2.1 웹표준 기반의 마크업	13
2.1.1 TABLE 기반 마크업	13
2.1.2 웹표준 기반의 마크업	13
2.1.3 웹표준 기반의 마크업 프로세스	13
3. 개발환경 설정	14
3.1 DTD 및 charset 설정	15
3.1.1 기본 설정	15
3.1.2 DTD 설정 참고 사항	15
3.1.3 CHARSET 설정 참고사항	16
3.1.4 자연어(LANG) 설정 참고 사항	16
3.2 폴더 생성	17
3.2.1 마크업 폴더 구조	17
3.2.2 폴더 생성 시 고려사항	17
작업 폴더를 생성할 때는 아래와 같은 사항을 고려합니다.	17
3.3 SVN 환경 설정	18
3.4 초기 파일 생성	18
3.4.1 HTML 파일 생성하기	21
3.4.2 CSS 파일 생성하기	21
3.5 기타 체크사항	21
3.5.1 IFRAME 적용시 유의사항	21
3.5.2 플래시 적용시 유의사항	21

4. 네이밍 규칙	22
4.1 공통 규칙	23
4.1.1 기본 규칙	23
4.1.2 조합 규칙	23
4.2 ID 및 Class	25
4.2.1 레이아웃	26
4.2.2 오브젝트	27
4.3 Image	29
4.3.1 기본 규칙	29
4.3.2 조합 규칙	29
4.4 File 및 Folder	31
4.4.1 기본 규칙	31
4.4.2 조합 규칙	31
5. 코드 작성 규칙	32
5.1 기본 규칙	33
5.1.1 완전기본규칙	33
5.2 HTML 기본 문법	34
5.2.1 HTML 코드의 정렬	34
5.2.2 HTML 주석 가이드	34
5.3 CSS 기본 문법	36
5.3.1 ID&CLASS 다중 선택자 사용 제한	36
5.3.2 CSS 코드의 정렬	36
5.3.3 CSS 속성 선언 순서	38
5.3.4 CSS 선택자 선언 순서	38
5.3.5 CSS 주석 가이드	39
5.4 HTML 엘리먼트 사용 규칙	40
5.5 핵 사용 규칙	43
5.6 접근성	47
5.6.1 대체텍스트를 제공해야함	47
5.6.2 특정환경에서만 제공되는 콘텐츠가 없도록 함	47
5.6.3 키보드만으로도 조작 가능해야 함	47
5.7 이미지 사용 규칙	50

웹표준, 웹접근성, 코드최적화, 생산량의 수치화

1. 웹표준 품질 측정 (수치화)

➤ 목표

- I. 코드 품질을 측정할 수 있는 수행과제 마련.
- II. 품질 측정을 통한 품질 수치화
- III. 품질 측정 전반에 대한 프로세스 마련
- IV. 수치화를 통한 코드 최적화, 단일화

2. 생산성 측정 (수치화)

➤ 목표

- I. 개발부서의 생산성을 측정하여 비교 가능하도록 수치화함.
- II. 측정된 수치를 통한 생산성 향상 방향 모색.

웹표준, 웹접근성, 코드최적화, 생산량의 수치화

1. 웹표준 품질 측정 (수치화)

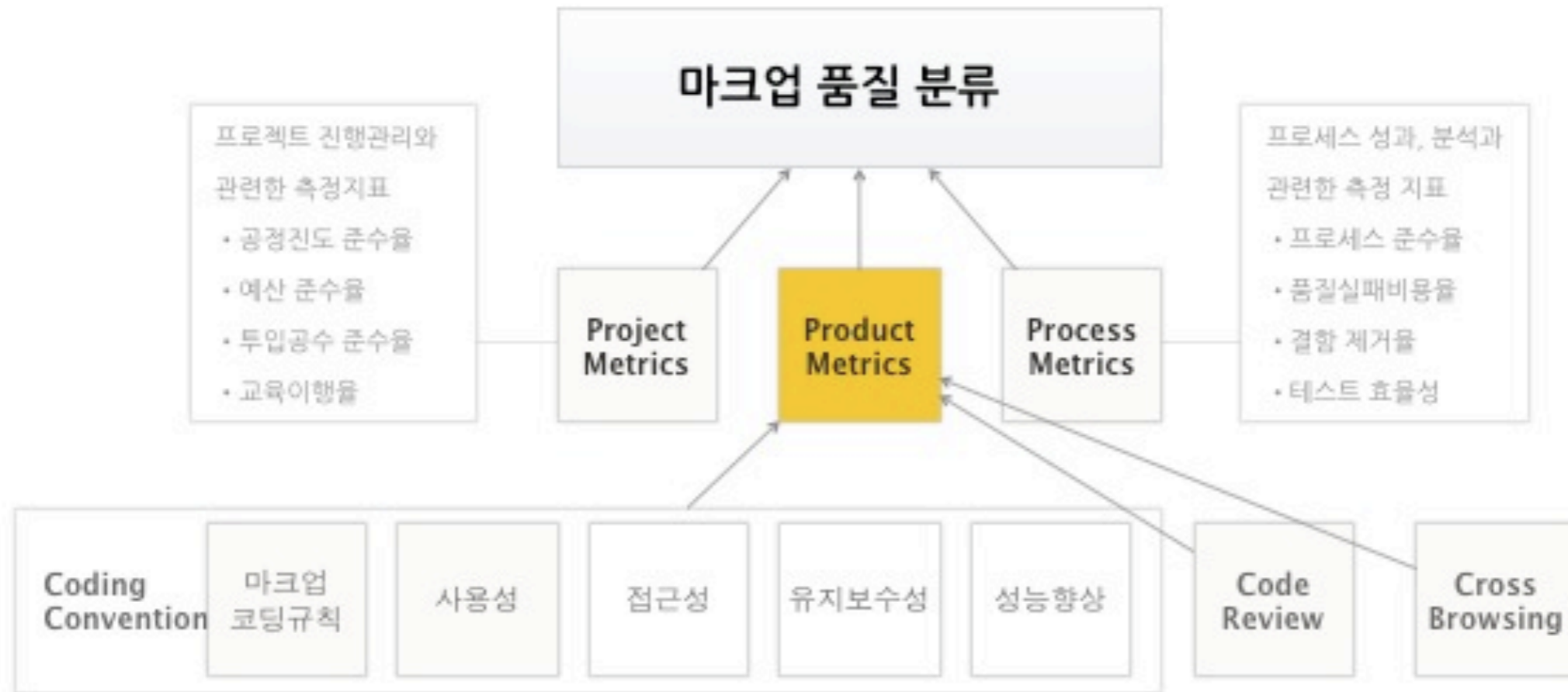
➤ 목표

- I. 코드 품질을 측정할 수 있는 수행과제 마련.
- II. 품질 측정을 통한 품질 수치화
- III. 품질 측정 전반에 대한 프로세스 마련
- IV. 수치화를 통한 코드 최적화, 단일화

2. 생산성 측정 (수치화)

➤ 목표

- I. 개발부서의 생산성을 측정하여 비교 가능하도록 수치화함.
- II. 측정된 수치를 통한 생산성 향상 방향 모색.



- 마크업 품질의 분류는 위와 같이 3개의 분류로 나눌 수 있음.
- 코드 품질 최적화, 단일화를 위해 3개의 분류 중 'Product Metrics'에 대한 프로세스 및 수치화를 정립.

4. QA

분류	측정지표	세부 항목
UI개발	LOC	LOC (Line of code)
	Coding Convention	코딩 규칙 준수율
		접근성 필수 항목 준수율
		접근성 권고 항목 준수율
		사용성 권고 항목 준수율
		유지보수성 권고 항목 준수율
	Code Review	코드 리뷰 수행율
	Cross Browsing	크로스 브라우징 기본 범위 검토율
		크로스 브라우징 확장 범위 검토율
	Static Code Analysis	W3C HTML Validation 통과 율
W3C CSS Validation 통과 율		
개발대응 유지보수	Defect Density(결함밀도)	Defect Density
		결함악성율(%)
		AR(After Release) Defect Density
		FFR(Fault Feedback Ratio)결함회귀 비율

4. QA

No	구분	항목명	통과율	해당없음 항목
1. 코딩 규칙 준수율				
1	1.1. 문서형(DTD) 선언	문서형(DTD) 선언 준수	100%	
2		모든 요소(Element), 속성(Properties), 값(Value)의 영문 소문자 사용 준수	100%	
3	1.2. 문자셋(charset)과 자연어 명시	문자셋(charset) 명시 준수	100%	
4		자연어 명시 준수	21%	
5	1.3. 의미에 맞는 태그로 마크업	h1~h6 요소의 순서 건너뛰기 금지	33%	5
6		레이아웃 목적의 Table 요소 사용 금지	80%	
7	1.4. Table의 바른 사용법	표를 설명하는 제목 요소 (h1~h6, caption, summary 중 한가지) 사용 준수	73%	
8		표의 머리글인 thead 요소 사용 준수	90%	
9		표의 본문인 tbody 요소 사용 준수	100%	
10		표의 머리글 셀인 th 요소 사용 준수	100%	

1. 개발명세 파악

- 1.1. 마크업 스펙 질의서
 - 1.1.1. 질의서 작성 가이드
- 1.2. 마크업 스펙 확정서

2. 일정산출

- 2.1. 목적
- 2.2. SO
- 2.3. 기획서 분석
- 2.4. 일정산출
- 2.5. 일정표 작성
- 2.6. 배포

3. 마크업 개발

- 3.1. 마크업 코드의 버전 관리
- 3.2. 마크업 코딩 컨벤션
 - 3.2.1. NULI
- 3.3. UIO
- 3.4. Pair Programming
- 3.5. HTS
 - 3.5.1. 개념
 - 3.5.2. 프로세스
 - 3.5.3. 모듈라이브러리
- 3.6. 마크업 메뉴얼
 - 3.6.1. 활용
- 3.7. 롤플레이닝
- 3.8. 릴리즈노트

4. 품질인증

- 4.1. 수치화
 - 4.1.1. 의미
 - 4.1.2. 적정지표
- 4.2. QP
 - 4.2.1. 개념
- 4.3. 체크리스트
- 4.4. 생산성 효율

End.