

'개발자의 글쓰기' 정리

작성일: 20.06.09

작성자: 도원진

1. 글쓰기 기본

1.1. 문장과 단락

1.1.1. 문장을 구조화하는 법

- 핵심 먼저, 그 뒤에 부가설명
 - 안좋은 예)
 - "색상 RGB값을 각각 사용하기 때문에 입력데이터는 3차원 벡터다."
 - 좋은 예)
 - "입력데이터는 3차원 벡터다. 색상 RGB값을 각각 사용하기 때문이다."
 - 핵심이 되는 주어가 항상 앞에 있다.
 - 부가설명은 맨 뒤로 배치한다.

1.1.2. 서술식/ 개조식/ 도식

- 서술식
 - ~~다
 - 개발문서
- 개조식(선호)
 - ~했음, 완료, 증대, ~함
 - 릴리즈문서, 장애보고서
- 도식
 - 표
 - 개조식에서 중복과 누락이 있을 수 있음.
 - 따라서 도식으로 각 항목의 공통, 차이를 분명히 표현한다.

1.2. 영어 단어 선택

1.2.1. 비슷한 듯 다른 듯

- 대표적인 반댓말
 - show vs hide
 - visible vs invisible
 - under vs over

- or under vs and over
- **stop, end, finish, pause, suspend, hold**
 - stop : 잠시 중단
 - end : 영원히 중단
 - finish : 끝장을 본 상태
 - pause : 순간만 중지
 - suspend : 다음단계로 넘어가서 시작직전에 중단
 - hold : 어떤 의도로 중단
- **get ,retrieve, fetch**
 - retrieve : 검색해서 가져오다
 - fetch : 현재 포인터가 다음 값으로 이동한 것을 가져오다
- **set, init, create, register**
 - set : 값 변경
 - init : 초기화
 - create: 새로운 틀을 만들 때
 - register: 이미 있는 틀에 등록할 때
- **change, modify, revise**
 - change : 단순 변경
 - modify : 잘못된 것을 바로 잡을 때
 - revise : 새로운 아이디어나 정보를 추가할 때

2. 개발 시간을 줄여주는 이름짓기와 주석쓰기

2.1. 네이밍 기본

2.1..1. 파스칼 표기법

첫 문자를 대문자로

예) Class명 : **S**tudentVO

2.1..2. 카멜표기법

첫 문자를 소문자로

예) 변수, 메소드, 함수, getNum();

2.1..3. 전체 소문자

예) 패키지, 모듈

2.2. 변수이름

2.2..1. 유의미한 i , 그렇지 못한 d

- i : index , integer를 뜻함

- i, j, k 가능
 - x, y 가능
- d 불가능(date, double 로 헷갈림)
 - day도 종류가 많다. 다음 단어로 의미를 밝혀서 사용할 것
 - someday
 - today
 - thisDay
 - finalDay

2.2..2. 복수형 표기

- s를 붙이는 것보다 Array, ListOf 라는 키워드를 넣는다.
 - checkUserNameArray
 - checkListOfUserName

2.2..3. 중요단어를 앞에 쓴다.

- 안좋은 예)
 - totalVisitor
 - totalRegister
 - totalBuyer
- 좋은 예)
 - visitorTotal
 - registerTotal
 - buyerTotal
 - salesOfThisMonthTotal
 - windowSizeMax

2.2..4. 함수 이름짓기

1. 요구사항 스토리를 문장단위로 분리한다
2. 함수하나 - 기능하나 로 문장을 더 쪼갬다.
3. 문장을 영어로 번역
4. 정관사(a, the), 소유격 삭제
5. check, get 등의 키워드를 사용
6. 몇몇 단어를 빼보는 시도를 해보고 없어도 이해할 수 있다면 더 삭제

2.2..5. 검색하기 쉬운 이름

- 범주화하여 일반화 -> 구체화한다.
 - 안 좋은 예)
 - SERVER_TIMEOUT
 - NO_RESULT
 - BAD_REQUEST
 - SERVER_ALLOWED_REQUEST
 - 좋은 예)
 - **ERROR_SERVER_TIMEOUT**
 - **ERROR_NO_RESULT**
 - **ERROR_BAD_REQUEST**
 - **ERROR_SERVER_ALLOWED_REQUESTS_EXCESS**
 - **user**

- `userBuyer`
- `uesrPayer`

2.2..6. 조합하기 쉽게 이름 짓는 법

- 문법 구문과 함께 조합해 이름짓는다

```
<h1 class="title"> 개발자의 글쓰기 </h1>
<h2 class="title"> 소프트웨어 엔지니어를 위한 테크니컬 라이팅 </h2>
<p class="title"> 개발자라면 이거 모르고 쓰지 마오! </p>
```

2.3. 다른 개발자를 위한 주석 쓰기

1. 이유를 알려줄 때
2. 새롭게 발견한 것
3. 예상 질문과 답
4. 할 일이나 주의, 개선 아이디어
5. 다른 사람에게 보완을 요청하는 것
6. 개발자의 속마음을 표현하는 것

3. 문서 보고서 , 장애보고서 쓰기

3.1. 체인지 로그 분류, 요약, 종합

양보다 질

- 1단계 : 선정하기
 - 1순위: 독자관심 多 , 개발자 노력 小
 - 2순위: 독자관심 多 , 개발자 노력 小
 - 3순위: 독자관심 小 , 개발자 노력 多
 - 4순위: 독자관심 小 , 개발자 노력 小
- 2단계 : 분류하기
 - 독자가 개발자인 경우
 - 기능단위로 분류
 - 예)
 - 새로운 기능
 - 개선된 기능
 - 오류수정
 - 독자가 일반사용자인 경우
 - 사용자관점의 스토리
 - 예)
 - 게임준비
 - 게임 중
 - 게임 종료
- 3단계 : 요약하기

- 2단계의 '서술식' 내용을 '개조식'으로 어미를 변경
- **4단계 : 종합하기**
 - 엘리베이터 스피치처럼 한 마디로 표현할 줄 알아야 함

사용자의 편리성 개선

- 게임방에서 더 빨리 입장
- 게임 결과 바로 확인

[세부 내용]

1. 게임 준비

-
-
-

2. 게임중

-
-
-

3. 게임종료

-
-
-

3.2. 릴리즈 문서는 문제해결 보고서처럼 쓰자

3.2.1. 다음 순서로 릴리즈 노트를 작성할 것.

- 문제와 문제점은 다르다.
- 문제 : 표면적으로 들어나는 현상
- 문제점 : 근본원인
- 해결책
- 후속계획

3.2.2. 면책사항과 더불어 종결어미 또한 중요

1. 필수

"~해야한다"

2. 권장

"~할 것을 권장한다"

3. 선택

"~하는 방법이 있다"

4. 개발 가이드 작성법

4.1. 설명, 묘사, 논증, 서사를 이용

- 범주, 용도, 특징 순으로 서비스의 개념을 설명할 것
 - 범주
 - 용도
 - 특징

4.1.1. 범주

- 경쟁사 비교우위표현
- 특정 사용자 타겟팅 표현
- 유명 개발사 브랜드네임 강조표현

4.1.2. 용도

- 범주의 핵심기능으로 기술
- 여러 기능들의 일반화

4.1.3. 특징

- 장점과 강점의 조합
- 장점
 - 주관적으로 잘하는 것(비교대상이 제품의 기능 중에서)
 - 객관적으로 잘하는 것(비교대상이 타기업)

4.2. 서사를 활용해 목차 만들기

1. 서사를 통해 일의 순서를 나열
2. 순서중 일반화 할 수 있는 항목들을 묶어 단계를 생성
3. 단계에서 목차를 만들기

5. 수주를 돕는 SI 제안서 쓰기

5.1. 제안요청서 분석

- 목표
- 하드웨어 구성도
- 소프트웨어 구성도
- 요구기능
- 위와 같은 요청을 하게 되는 배경과 상황이 정답처럼 들어있음.

5.2. 논리적 완결성

- 고객은 제안서를 소설처럼 처음부터 읽지 않는다.
- 앞 페이지에서 참고해서 읽게 만들지 말것.
- 예) 시스템 구성도
 - 고객의 요구
 - 고려사항
 - 구성전략

- 근거
- 위의 조건들을 함께 명시해야 함.
- 앞 페이지에서 참고해서 읽게 만들지 말것.

5.3. 고객의 문제인식과 개발사의 해결능력

5.3.1. CASE 1) 문제인식高, 해결능력 高

- 개발사의 해결 능력이 출중해서 그것들을 나열해도 고객의 눈에 들어오지 않음.
- 경쟁사와 비교자료를 제시하여 대조된 예상치를 보여준다.

5.3.2. CASE 2) 문제인식高, 해결능력 低

- 억지로 기술구현을 하려고 하지 않는다.
- 다른 대안책을 내놓는다.
- 기술이 낮아도 더 큰 인력이 소요되더라도 그 방법을 제시하고 논리적으로 더 좋은 대안임을 설명

5.3.3. CASE 3) 문제인식低, 해결능력 高

- 고객이 높은 문제인식을 갖을 수 있도록 여러 자료로 설득한다.
- 높은 기술력을 가지고 있기에 타 경쟁사는 신경쓰지 않는다.
- 오로지 고객의 문제인식을 바꾸는데 노력한다.

5.3.4. CASE 4) 문제인식低, 해결능력 低

- 경쟁사의 전략을 확인해서 대처할 것

5.4 고객은 변덕쟁이

5.4.1. 요구사항을 분석하지 말고 제시하라

- 최초 요구사항은 추상적이다.
- 고객 자신이 무엇을 원하는지 모른다.
- 따라서, 개발자는 그 요구사항을 구체화하고 역으로 제시할 줄 알아야함.

5.4.2. 변화하는 요구사항에 대비할 것

- 개발프로세스의 각 단계를 짧게 잡고 프로토 타입을 미리 만들고 고객에게 검수를 받는다.

6. 기술 블로그 쉽게 쓰고 운영하기

6.1. 주제의식을 버리고 '소재 의식'으로 쓰자.

- 주제의식
 - 민족, 권선징악, 교훈
- 소재의식
 - 오직 관심사에 집중함
 - 주제가 없다
 - 상황에 몰두

6.2. 독자를 고려하지 않은 글

- 기술블로그의 독자는 다양한 연령대, 다양한 스펙폭을 가지고 있음
- 모두를 아우르는 글쓰기는 불가능하다.
- 낯선 용어에 대해서 설명이 있는 링크첨부까지는 예의상 넣어준다.

6.3. 재미있는 글을 쓰자

- 비유, 경험담, 공감을 일으키는 소재를 함께 섞어서 기술한다.

6.4. 블로그 글 종류 - 저,술,편,집

6.4.1. 저

- 개발기, 도입기, 적용기
- 직접 경험하고 실험한 과정
- 목차를 잡고 본문부터 작성한다.
- 제목은 마무리할 때 작성

6.4.2 술

- 기술 소개, 용어분석, 에러해결방법
- 기술을 분석해 의미풀이, 해석한 글
- 원전을 비교>실험>풀이

6.4.3 편

- 프로그램 설치, 설정, 튜토리얼, 세미나 후기, 책리뷰
- 기술을 분석해 의미풀이, 해석한 글
- 순서를 요약하는 글쓰기
- 상위 목차, 하위목차 순으로 글의 구조를 순서에 유의하며 적는다

6.4.4 집

- 명령어 모음, 팁, OO규칙
- 기술을 분석해 의미풀이, 해석한 글
- 흩어진 여러권의 자료를 한데 모아 정리한다.