소프트웨어 글로벌화 체크리스트

2019. 2.

정보통신산업진흥원

별첨 3	글로벌화 체크리스트

A. 기본 항목

A01	목표 언어와 지역을 식별한다.	
A02	지역적 요구사항을 수집한다.	
A03	기업, 제품, 적용 기술 등에 대한 주요 용어를 식별하고 그에 대한 용어사전을 작성한다.	
A04	글로벌하게 사용될 수 있는 디자인 스타일과 배치에 대한 가이드라인을 정의하여 템플릿으로 작성한다.	
A05	CSS 등과 같은 별도의 스타일 시트를 사용한다.	
A06	소스코드, 빌드, 콘텐츠 및 그 밖의 자료를 전역/지역 용도로 나누어 준비한다.	
A07	코드로부터 텍스트와 그 밖의 자원을 분리한다.	
A08	비즈니스 로직을 사용자 인터페이스로부터 분리한다.	
A09	클래스명, 프로그램 변수명, 메타데이터 등에 각 객체의 용도를 반영하는 논리적 이름을 사용한다	
A10	"right", "left"가 포함된 변수나 메타데이터 명을 생성하지 않는다. 단, before, after는 포함 가능하다.	
A11	국제적 요구사항을 만족하는 메타데이터를 설계한다. (예를 들어, 메타데이터 명을 그대로 이용자에게 노출하지 말고 각 지역시장에 적합하도록 번역/대치하여 사용한다.)	
A12	웹 서비스의 경우 지역적 도메인을 지원한다. (.jp, uk, .sg 등)	

B. 다지인과 배치

B01	사용자 인터페이스와 보고서를 최대한 단순하게 유지한다.	
B02	스크린과 보고서를 설계할 때 번역으로 인해 텍스트가 확장되는 경우를 대비한 공간을 고려한다.	
B03	스크린과 보고서를 설계할 때 더 큰 폰트를 필요로 하는 언어로 번역되는 경우에 대해 수직적 확장을 고려한다.	
B04	대부분의 OS, 디스플레이, 프린터 등에서 지원되는 일반적인 폰트를 사용한다.	
B05	웹 서비스의 경우, HTML과 CSS에는 대비책으로 범용 폰트를 포함한다.	
B06	폰트명, 폰트 패밀리 및 사이즈의 현지화를 지원한다.	
B07	머리글, 바닥글, 페이지 번호 삽입 등은 현지화가 적용되면 함께 변경되므로, 자동적으로 현지화된 버전이 반영되도록 한다.	
B08	지역에 따라 다양한 디스플레이, 프린터 등에 대한 폼 항목/양식을 지원한다. (스크린, 용지 크기, 해상도, 레이아웃, 출력 비용 등)	
B09	글로벌하게 사용 가능한 그래픽스를 설계한다. 예를 들어, 심볼을 설계할 때 추상화 기법을 사용하여 단순화하거나, 또는 일반화 기법을 사용하여 전 세계 어느 나라에서도 사용될 수 있도록 보편화할 수 있다.	
B10	제품의 특성에 따라 언어 선택을 어떠한 방식으로 제공할지 고려한다. 예를 들어, 해당 소프트웨어의 첫 페이지, 환영 페이지, 또는 별도의 메뉴에서 한번 이상 제공할지 여부 등을 결정한다.	
B11	한 국가에서 여러 언어를 사용하는 경우가 많으므로 언어 선택을 위해 국기를 사용하지 않는다.	
B12	다중 언어/문화권의 이용자를 배려하기 위해 기본 언어 선택과 별도로 이용자들이 원하는 데이터 양식을 선택할 수 있는 이용자 인터페이스를 제공한다.	
B13	위치와 크기는 언어 선택에 따라 자동으로 최적화하거나, 혹은 이용자가 선택적으로 변경할 수 있도록 적절한 이용자 인터페이스를 제공한다.	

C. 그래픽스

C01	문화를 구체적으로 반영하는 아이콘과 그래픽스는 피한다. (예를 들어, 사람, 신체일부/몸짓, 나이/성별/민족/관계, 의복/제복, 동물, 유머, 다의어, 동음이의어 등을 이용한 말장난/수수께끼, 다양한 의미를 갖는 그래픽스, 특정 문화에 대한 객체 또는 모양, 은유/비유, 스포츠 용어, 종교적 심볼, 세계지도 등 전체 행성을 보여주지 않는 지구의 이미지, 계절, 색상, 색상 조합 등)	
C02	레이아웃, 순서, 성향 등이 읽기/쓰기 방향에 따라 변경될 수 있도록 고려한다. (예를 들어, 중동에서는 오른쪽에서 왼쪽방향으로 글을 쓰고 가장 중요한 정보는 상위 오른쪽에 배치된다.)	
C03	아이콘, 이미지, 그래픽스 등에 임베디드된 텍스트의 사용을 지양한다.	
C04	그래픽스 근처의 호출, 텍스트 등이 번역되었을 때 확장 가능하도록 여분의 공간을 고려한다.	
C05	그래픽스에 임베디드된 텍스트를 사용해야 한다면 그래픽스와 텍스트를 각각 저장하여 현지화 프로그램이 각각 호출하여 조합할 수 있도록 한다.	
C06	불필요한 스크린 캡쳐의 사용을 지양한다.	
C07	스크린을 캡쳐하기 위해 사용되는 프로시져, 컴퓨터 환경 및 소프트웨어 세팅을 문서화한다.	

D. 자원 파일, 현지화용 콘텐츠

D01	사용자 인터페이스 텍스트를 코드로부터 분리한다.	
D02	번역 가능한 텍스트로부터 번역되지 않아야 하는 텍스트를 명확히 식별하거나 분리한다.	
D03	그래픽스, 오디오, 비디오, 그 밖의 미디어를 코드로부터 분리한다.	
D04	스크린과 보고서 관련 항목의 각 위치와 크기 정보는 코드로부터 분리한다.	
D05	폰트와 폰트 크기는 현지화 프로그램과 이용자가 변경할 수 있도록 한다.	
D06	"주어부:술부"의 형태, 또는 완성된 문장의 형태를 사용한다.	
D07	문자열 번역 후 조합 시 발생 가능한 이슈를 줄이기 위해, 단어 조각들을 연결하여 문장을 생성하기보다 최대한 많은 텍스트를 하나의 단위로 구성한다.	
D08	각 문장에 삽입되는 변수의 개수를 최소화한다. 문자열 리소스 한개 당 2개 이상의 변수가 대치되는 것을 지양한다.	
D09	번역 대상 문자열에 사용되는 변수들을 문서화하여 효율을 높인다.	
D10	번역 대상 문자열이 확장될 수 있도록 여분의 공간을 확보한다.	
D11	자원 파일이나 메시지 카탈로그에 문자열이 추가될 때 새로운 식별자를 생성한다.	
D12	소스 문자열의 텍스트와 동일한 문자열 식별자를 사용하지 않는다.	
D13	각 문자열에 대해 독립적인 인스턴스를 리소스 파일이나 메시지 카탈로그에 추가하여 같은 문자열이 다른 의미로 번역되는 경우를 지원한다. (예를 들어, orange가 색상 혹은 과일로 해석되는 경우)	
D14	에러 메시지는 유일한 식별자로 출력하여 각자 언어별 번역결과가 디스플레이 되도록 한다.	

E. 텍스트 처리 프로그래밍

E01	ASCII, ISO 8859-1 등 특정 코드 페이지를 추정하지 않는다.	
E02	한 문자가 1 바이트라고 추정하지 않는다. 즉, 문자열 길이가 문자 개수에 비례한다고 추정하지 않는다.	
E03	문자 개수를 세거나 문자열을 인덱싱할 때 단위를 추정하지 않는다.	
E04	각 문자 대신 개별 바이트를 처리하지 않는다.	
E05	복수 바이트 문자들을 쪼개지 말고 전체 문자를 대상으로 처리한다.	
E06	문자열이나 버퍼를 절단할 때 복수 바이트 문자들을 쪼개지 않는다.	
E07	포인터로 텍스트를 참조하는 API의 경우 문자의 첫 번째 바이트부터 참조한다.	
E08	유니코드로 구성되지 않은 문자열에 대한 알고리즘을 설계할 때, 문자열 처리 순서가 전반->후반인지, 후반->전반인지 명확하지 않은 경우엔 전자를 따른다.	
E09	한 번의 타자가 문자 한 개를 생성한다고 추정하지 않는다. 예를 들어, 액센트/톤 표시/발음구별부호 등을 수반한 문자는 2번 이상의 타자를 필요로 한다.	
E10	한 개 문자를 문자열에 추가하고 나면 커서가 한 칸 앞으로 나간다고 추정하지 않는다. 예를 들어, 액센트/톤 표시/발음구별부호 등은 베이스 문자의 앞/뒤/전/후 등에 위치할 수 있다.	
E11	커서가 오른쪽으로, 일직선으로 움직인다고 추정하지 않는다. 언어에 따라, 좌->우, 상->하 방향, 또는 특정 문자에 따라 양방향(좌 또는 우)으로 기록되는 경우도 있다.	
E12	디스플레이 혹은 인쇄되는 문자열의 길이를 구하기 위해 전체 문자 내용을 참조한다. 언어에 따라 인접 문자에 따라 모양이 변하는 경우도 있으므로 바이트 수를 세지 말고, 전체 문자열이 주어진 다음 길이를 구하는 것이 안전하다.	
E13	언어에 따라 문자의 개수가 문자열의 길이와 비례하지 않음을 고려하여 문자열에 대해 충분한 메모리를 효율적으로 할당한다.	

E14	개발자가 직접 포인터를 다루기 보다는 GetNext, GetPrev 모델을 사용한다.	
E15	하드 코딩된 코드 포인트를 문자로 변환하는 함수의 사용을 피함으로써, 도중에 인코딩 방식이 바뀌거나 값이 인코딩 변환을 통해 넘어갈 때 발생 가능한 문제를 예방한다. 이러한 방식이 반드시 필요하다면, 베이스 문자로 유니코드로 사용한다.	
E16	개발자가 개인적인 활용을 위한 신호 또는 모니터링 등의 용도로 특정 문자들을 문자열에 할당하지 않는다. 인코딩 방식이 바뀌거나, 로직이 꼬여 이러한 문자들이 텍스트에 포함되는 경우 문제가 발생할 수 있다.	
E17	가장 큰 문자열 값을 구하기 위한 문자열 생성을 지양한다. 인코딩 방식에 따라 가장 큰 문자열 값이 달라지므로 문제가 될 수 있다.	
E18	단어가 공백, 쉼표, 마침표 또는 기타문장 부호에 의해 분리된다고 추정하지 않는다.	
E19	문자의 속성(alphabetic, numeric, punctuation 등)은 언어와 문맥에 따라 다양할 수 있으므로 국제화 라이브러리 함수를 사용해서 검사한다.	
E20	모든 문자가 서드파티 제품에서도 같은 의미를 가진다고 추정하지 말고 모든 관련 컴포넌트를 점검하고 그 의미를 명확히 해야 한다.	
E21	한 서버에 연결된 모든 이용자가 같은 언어를 공유한다고 추정하지 않는다. 언어 및 데이터 양식 부분을 다시 한번 점검한다.	
E22	언어가 변경되면 세션이나 프로세스의 정지 및 재시작을 추정하지 않고 재점검한다.	
E23	모든 프로세스가 알파벳순과 같은 단일 정렬 체계를 사용한다고 추정하지 않는다.	
E24	내부적으로 사용하기 위한 정규화 양식을 정의한다.	
E25	일반적으로, API를 호출하기 전에 전체 문자열을 구성한다.	

E26	API를 설계할 때 인수로써 한개 문자가 아닌 문자열을 사용한다.	
E27	API는 언제나 에러가 발생할 수 있다는 가정 하에, 에러 발생 시 적합한 진단 프로그램이 호출되도록 한다.	
E28	대소문자를 특별히 구분해야 하는 경우가 아니면, 문자열 처리 시 소문자로 변환하여 사용하는 것이 안전하다.	
E29	특정 키보드 레이아웃을 추정하지 않는다.	
E30	모든 키보드에 단축키가 유효하다고 추정하지 않는다.	
E31	단축키에 대한 번역 방침을 정한다.	

F. 문자 인코딩 변환

F01	내부적으로 합당한 표준 인코딩을 선택한다.	
F02	각 플랫폼이나 시스템 컴포넌트, 프로세스, 프로토콜, 장치, 데이터베이스, 또는 파일 포맷에 사용되는 문자 인코딩을 식별한다.	
F03	다른 인코딩을 사용하는 컴포넌트 간의 적합한 인코딩 변환을 보장한다.	
F04	인코딩 변환의 상황을 파악하고 이슈가 되는 경우에 대해 중재할 준비를 갖춘다.	
F05	특정 플랫폼에서 텍스트를 교환할 때, 해당 플랫폼과 같은 변환 테이블을 사용하도록 한다.	

G. 로케일 처리

G01	이용자 텍스트를 파싱하거나 포맷팅할 때 설정된 로케일에 따른다.	
G02	달력 세팅, 일/월/년도의 명칭 등을 포함한 날짜 포맷을 현지화 한다.	
G03	날짜와 시간을 이용한 산술 계산 시 현지화된 기준을 고려한다.	
G04	시간 포맷을 현지화 한다.	
G05	지역 표준시간대(time zone)는 설정 가능하도록 구성한다. 한 나라에 여러 시간대가 포함될 수 있으므로 표준시간대는 국가 코드로 식별될 수 없다.	
G06	써머타임에 대한 설정 및 변경이 가능하도록 구성한다.	
G07	통화 (심볼, 정밀도, 반올림 규칙, 폭, 세금, 화폐 등) 관련 정보를 현지화 한다.	
G08	숫자 포맷 (십진법, 숫자 그룹 분리자, 개별 숫자 속성 등) 관련 정보를 현지화 한다.	
G09	폰트 (이름, 크기 등) 관련 정보를 현지화한다.	
G10	주소 포맷 (우편 번호, 지역 단위 등) 관련 정보를 현지화한다.	
G11	이름 포맷 (이름의 개수, 성(가족 이름)의 순서, 칭호 등) 관련 정보를 현지화 한다.	
G12	전화 번호, 운전면허번호, 여권번호, 자동차 번호 등 각종 식별자(ID) 관련 정보를 현지화 한다.	
G13	각종 측정 단위 관련 정보를 현지화 한다.	
G14	종이 크기 관련 정보를 현지화 한다.	
G15	지역 문화에 따라 의미가 부여된 색상 관련 정보를 현지화 한다.	
G16	정렬 순서 관련 정보를 현지화 한다.	
G17	특정 리스트가 번역된 이후에 배치되는 순서가 기록되어야 한다.	
G18	대문자 사용 관련 규칙 또는 정보를 현지화 한다. 언어에 따른 대소문자 구분 여부, 대소문자 간의 관계, 대문자를 적용하는 문법상의 규칙 등 고려한다.	
G19	쓰는 방식 (강조, 인용, 하이픈 연결, 적합성 등) 관련 정보를 현지화 한다.	
G20	쓰기 방향 (좌->우, 우->좌, 상->하) 관련 정보를 현지화 한다.	

H. 웹 국제화

H01	표 대상 콘텐츠에 적합한 문자 인코딩을 사용한다.	
H02	올바른 문자 인코딩을 시스템 구성 소스 코드의 적합한 곳에 표시한다. (각종 웹 페이지-HTML, CSS, XML 등, 클라이언트/서버 페이지 등)	
H03	서버가 수용할 수 있도록 문자 인코딩을 명세 한다.	
H04	콘텐츠 내에 사용된 언어명을 표기한다. 예를 들어, HTML이라면 lang속성, XML이라면 xml:lang 속성을 이용한다.	
H05	우->좌 방향으로 표기되는 언어가 포함되는 HTML이라면 DIR, BDO 등의 속성을 사용한다.	

I. 텍스트 저작

I01	특정 수준의 읽기 수준을 목표로 한다.	
I02	언어 스타일에 대한 가이드라인을 설정한다.	
I03	콘텐츠에 대해 선택된 분위기, 목표 국가/지역 이용자에 대해 전달하고자 하는 메시지, 이유 등을 식별하고 기록한다.	
I04	같은 단어는 텍스트를 재사용하는 등 사용된 단어의 개수를 최소화 한다.	
I05	현지화 작업과 관련된 엔지니어링 업무를 계획한다.	
I06	사용자 인터페이스와 해당 문서 간에 번역의 일관성을 유지한다.	
I07	명확하고 구체적인 언어를 사용한다.	
I08	문장은 짧게 유지한다. 과도한 쉼표의 사용 및 복합 구분의 사용은 피한다.	
I09	의미 전달의 모호성을 예방하기 위해 직접 또는 간접 객체에 대한 번역을 명확하게 처리한다.	
I10	속어, 은어, 어려운 용어, 숙어나 관용구 등의 사용은 지양한다.	
I11	두문자어, 축약어 등은 정의한다.	
I12	관계대명사는 포함한다. (who, that, which 등)	
I13	문장과 질문은 가급적 긍정문으로 표기한다.	
I14	특정 문화에 대한 구체적인 내용을 참조하는 표현은 가급적 지양한다.	
I15	문화적인 참조를 포함해야 한다면 압축/요약해서 해당 내용이 현지화 프로그램에 의해 손쉽게 발견되고 대치될 수 있도록 한다.	
I16	목표 지역에 적합한 데이터 포맷을 사용한다.	
I17	목표 지역 이용자의 요구에 따라 번역 순서를 효율적으로 재정렬할 수 있도록 번역 대상에 대해 내용 영역별로 구체적인 계획을 세운다.	
I18	효과적인 번역을 위해 동명사의 사용으로 인한 모호성을 지양한다.	
I19	효과적인 번역을 위해 가급적 새로운 단어를 만들지 않는다.	
I20	효과적인 번역을 위해 조동사의 사용으로 인한 모호성을 지양한다.	
I21	특정 문화권에서 적합하지 않은 스타일의 사용을 지양한다.	

J. 도구

J01	변화관리를 준비한다.	
J02	현지화 프로그램은 번역 메모리 (Translation Memory, TM)을 지원하는 도구를 사용한다.	
J03	국제화를 지원하는 도구를 사용한다.	