DIM0436
Especificações informais

Richard Bonichon

20140729

Sumário

Uma citação

My experience has shown that many people find it hard to make their design ideas precise. They are willing to express their ideas in loose, general terms, but are unwilling to express them with the precision needed to make them into patterns. Above all, they are unwilling to express them as abstract spatial relations among well-defined spatial parts. I have also found that people aren't always very good at it; it is hard to do...
Christopher Alexander (1979), The Timeless Way of Building.

Estilos de especificação

Nível formal

Informal em linguagem natural

Estilos de especificação

Nível formal

Informal em linguagem natural

Formal sintaxe e semântica formalmente definidas.

Estilos de especificação

Nível formal

Informal	em linguagem natural
Formal	sintaxe e semântica formalmente definidas.
Semi-formal	em geral apenas a sintaxe (e parte da semântica) formalmente definida.

Dicas de redação

- Evitar:

Dicas de redação

- Evitar:
  - frases longas

Dicas de redação

- Evitar:
  - frases longas
  - excesso de pronomes

Dicas de redação

- Evitar:
  - frases longas
  - excesso de pronomes
  - nomes diferentes para o mesmo conceito

Dicas de redação

- Evitar:
  - frases longas
  - excesso de pronomes
  - nomes diferentes para o mesmo conceito
  - homônimos (mesmo nome para conceitos diferentes)

Dicas de redação

- Evitar:
  - frases longas
  - excesso de pronomes
  - nomes diferentes para o mesmo conceito
  - homônimos (mesmo nome para conceitos diferentes)
  - termos vagos: alguns, às vezes, frequentemente, ...

Dicas de redação

- Evitar:
  - frases longas
  - excesso de pronomes
  - nomes diferentes para o mesmo conceito
  - homônimos (mesmo nome para conceitos diferentes)
  - termos vagos: alguns, às vezes, frequentemente, ...
- Usar enumerações

Dicas de redação

- Evitar:
  - frases longas
  - excesso de pronomes
  - nomes diferentes para o mesmo conceito
  - homônimos (mesmo nome para conceitos diferentes)
  - termos vagos: alguns, às vezes, frequentemente, ...
- Usar enumerações
- Estruturar o documento

Dicas de redação

- Evitar:
  - frases longas
  - excesso de pronomes
  - nomes diferentes para o mesmo conceito
  - homônimos (mesmo nome para conceitos diferentes)
  - termos vagos: alguns, às vezes, frequentemente, ...
- Usar enumerações
- Estruturar o documento
- Permitir apenas algumas redundâncias

Dicas de redação

- Evitar:
  - frases longas
  - excesso de pronomes
  - nomes diferentes para o mesmo conceito
  - homônimos (mesmo nome para conceitos diferentes)
  - termos vagos: alguns, às vezes, frequentemente, ...
- Usar enumerações
- Estruturar o documento
- Permitir apenas algumas redundâncias
- Uso de siglas

Dicas de redação

- Evitar:
  - frases longas
  - excesso de pronomes
  - nomes diferentes para o mesmo conceito
  - homônimos (mesmo nome para conceitos diferentes)
  - termos vagos: alguns, às vezes, frequentemente, ...
- Usar enumerações
- Estruturar o documento
- Permitir apenas algumas redundâncias
- Uso de siglas
  - a primeira ocorrência SEMPRE acompanhada da forma por extenso (e glossários)

Exemplos

Exemplo (Identificação do operador)

A identificação do operador consiste dos seus nome e senha; a senha consiste de 6 dígitos. Ela deve ser mostrada no painel e registrada no arquivo de login quando o operador entra no sistema.

Exemplo (Mensagem)

A mensagem deve ser triplicada. As 3 cópias devem ser enviadas através de 3 canais físicos diferentes. O receptor aceita a mensagem baseado em uma votação com maioria (2 em 3).

Características

Vantagens

- Uso da língua natural

Observação

Definições de termos são muito importantes

Características

Vantagens

- Uso da língua natural
- Conhecimento geral, língua usada para comunicação

Observação

Definições de termos são muito importantes

Características

Vantagens

- Uso da língua natural
- Conhecimento geral, língua usada para comunicação
- Sem notação específica

Observação

Definições de termos são muito importantes

Características

Vantagens

- Uso da língua natural
- Conhecimento geral, língua usada para comunicação
- Sem notação específica
- Sem restrições, alem da sintaxe e semântica

Observação

Definições de termos são muito importantes

Desvantagens

Contra

- Uso da língua natural
- Discórdia!
- Nível de informalidade alto
- Contradições
- Ruido

Impedimento

Definição

- A player is in an offside position if: he is nearer to his opponents' goal line than both the ball and the second last opponent
- A player is not in an offside position if: he is in his own half of the field of play or he is level with the second last opponent or he is level with the last two opponents
- A player in an offside position is only penalised if, at the moment the ball touches or is played by one of his team, he is, in the opinion of the referee, involved in active play by: interfering with play or interfering with an opponent or gaining an advantage by being in that position
- There is no offside offence if a player receives the ball directly from: a goal kick or a throw-in or a corner kick.

Jogo da vida

Regras I

1. If a cell is currently dead and has 3 live neighbors, it resurrects and becomes live again;
2. If a cell is currently alive and has two or three live neigbhors, it remains alive;
3. Otherwise, it dies

Jogo da vida

Regras I

1. If a cell is currently dead and has 3 live neighbors, it resurrects and becomes live again;
2. If a cell is currently alive and has two or three live neigbhors, it remains alive;
3. Otherwise, it dies

Regras II

1. Any live cell with fewer than two live neighbours dies, as if caused by under-population.
2. Any live cell with two or three live neighbours lives on to the next generation.
3. Any live cell with more than three live neighbours dies, as if by overcrowding.
4. Any dead cell with exactly three live neighbours becomes a live cell, as if by reproduction.

Tabela de dispersão/hash

Função de *hash h*

Colisões entre chaves devem ser raras. I.e. dadas chaves k_1 e k_2, a função de *hash* h deve calcular $h(k_1)$ $h(k_2)$ diferentes na maioria do tempo.

Definição (Definição do padrão C++)

1. The value returned by h(k) shall depend only on the argument k.
2. For two different values k1 and k2, the probability that h(k1) and h(k2) " compare equal" should be very small.

Observações

Regra 1

The value returned by h(k) shall depend only on the argument k.

- h é determinista, i.e. não pode depender de uma variável aleatória.
- visita das chaves sempre na mesma ordem
- Se $h(k_1) = h(k_2)$, é para todos os programas e todas as execuções.

Observações

Regra 1

The value returned by h(k) shall depend only on the argument k.

- h é determinista, i.e. não pode depender de uma variável aleatória.
- visita das chaves sempre na mesma ordem
- Se $h(k_1) = h(k_2)$, é para todos os programas e todas as execuções.

Regra 2

For two different values k1 and k2, the probability that h(k1) and h(k2) " compare equal" should be very small.

- no modelo aleatório, \$h(k_1), h(k_2) são constantes.
- o que isto significa ?
  - se escolher aleatoriamente k_1 e k_2, a probabilidade de colisão é baixa ?
    - se for o caso é uma especificação fraca, i.e. podemos *hash* (picar?) *strings* para a primeira letra delas . . .

O jogo

Regras originais

Para contar os pontos:

1. Black key pegs are placed by the Codemaker in any of the key peg holes

for every code peg placed by the Codebreaker which has the colour and is in exactly the same position as one of the code pegs behind the shield.

1. White key pegs are placed by the Codemaker in any of the key peg holes when any hidden code peg behind the shield matches the Codebreaker's code pegs in colour only but not in position.

É suficiente ?

Um extrato do jogo

Solução	B	Y	G	B
Proposta	Y	W	Y	Y

É suficiente ?

Um extrato do jogo

Solução	B Y G B
Proposta	Y W Y Y

Problema

A proposta tem 3 pinos Y, todos eles correspondem ao pino Y da solução. Eu preciso colocar 3 pinos brancos. Ou só um ?

É suficiente ?

Um extrato do jogo

Solução	B Y G B
Proposta	Y W Y Y

Problema

A proposta tem 3 pinos Y, todos eles correspondem ao pino Y da solução. Eu preciso colocar 3 pinos brancos. Ou só um ?

Escpecificação adicional (original)

Example: If one red code peg is behind the shield and the Codebreaker places two red code pegs in the [sic] position, ONE white key peg is used.

Descrição declarativa

Hipóteses

- Sequências s (solução) e p (proposta) de n cores são dadas.
- Especificar a marcação = especificar funções $b(s, p)$ e $w(s, p)$ dado o

número de cavilhas pretas e brancas

Especificação

- Uma correspondência forte entre s e p e um par (s_i, p_i) tal que $s_i = pi$
- Dois pares (s_i, pj) e (s_k, pl) são disjuntos se e somente se $i \neq k \wedge l \neq k$
- Uma correspondência fraca entre s e p e um par (s_i, p_j) disjunto de qualquer par forte e tal que $s_i = pj$
- $b(s, p)$ é o número de correspondências fortes entre s e p
- Lemma *Todos os conjuntos máximos de correspondências fracas mutualmente disjuntas têm o mesmo tamanho*
- $w(s, p)$ é o tamanho de um conjunto máximo de correspondências fracas mutualmente disjuntas.

Descrição operacional informal

1. For each location where the code and probe have a peg of the same colour, remove both, and place a black scoring peg.
2. For each code peg and probe peg that have the same colour, remove both, and place a white scoring peg.

Uma descrição operacional (em C)

```c
typedef enum {white, green, yellow, red, black, blue} colour;
typedef enum {false, true} bool;
typedef struct {int blacks; int whites;} score;

score fscore(colour code[], colour probe[], int n) {
  score s = {0, 0};
  bool cc[n];  // characteristic function for subset of code[]
  bool pp[n];  // characteristic function for subset of probe[]
  for(int i=0; i < n; i++) {
    if (code[i] == probe[i]) {
      s.blacks++;
      cc[i] = pp[i] = false; // this pair is counted; remove it
    }
    else cc[i] = pp[i] = true // check this pair later for white scoring peg
  }
  for(int i=0; i < n; i++) {
    for(int j=0; j < n; j++) {
      if (cc[i] == true && pp[j] == true // both pegs present
          && code[i] == probe[j]) {
        s.whites++;
        cc[i] = pp[j] = false; // remove both pegs
      }
    }
  }
  return s;
}
```

Conclusão

Observação

Nenhum jogador tem problema com o jeito de contar os pontos

Discussão

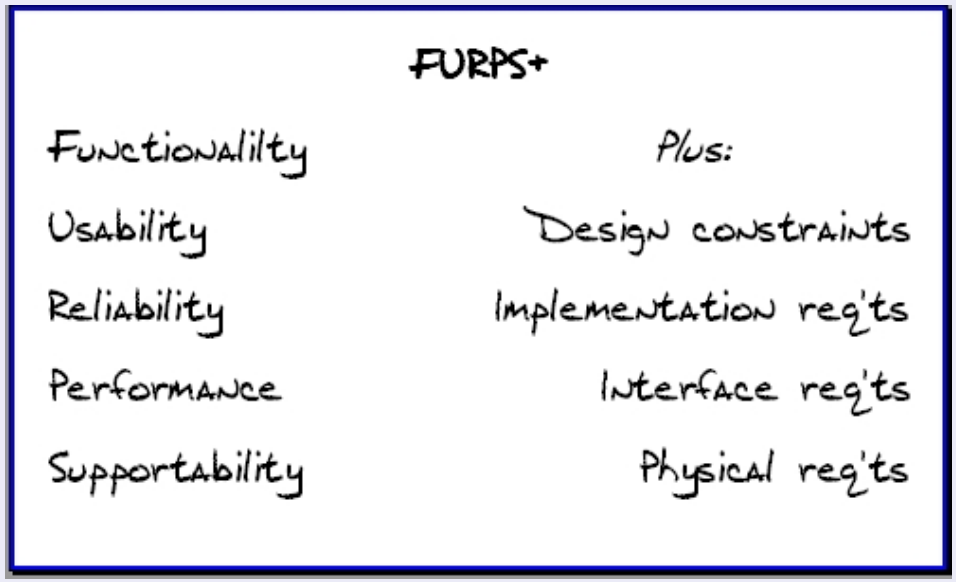
Qual é a melhor ?

FURPS

Definição (FURPS+)

FURPS+ é um sistema elaborado por HP para classificar requisitos.
O acrônimo representa categorias que podem ser usadas na definição de requisitos, assim como representa atributos de qualidade de Software.

FURPS+



Categorias FURPS

Funcionalidades

Aspectos principais do produto. Por exemplo, como processar ordens ou controlar o estoque de uma loja virtual. Não precisam ser tão especificos: capacidades de impressão são também requisitos funcionais.

Usabilidade

Estética, consistência da interface usuário.

Confiabilidade

Integridade, conformidade e interoperabilidade do software.
Os requisitos a serem considerados são: freqüência e gravidade de falha; possibilidade de recuperação; possibilidade de previsão; exatidão dos cálculos ; tempo médio entre falhas, ...

Categorias FURPS

Desempenho

Tempo de resposta, consumo de memória, utilização da CPU, capacidade de carga, disponibilidade da aplicação

Suportabilidade

Testabilidade, adaptabilidade, manutenibilidade, compatibilidade, configurabilidade, instalabilidade, escalabilidade, localizabilidade entre outros.

Categorias +

Desenho

especifica ou restringe o design de um sistema: linguagens de programação, processo de software, uso de ferramentas de desenvolvimento, biblioteca de classes, ...

Implementação

especifica ou restringe o código ou a construção de um sistema: padrões obrigatórios; linguagens de implementação; políticas de integridade de banco de dados; limites de recursos; ambientes operacionais...

Físico

especifica uma limitação física pelo hardware utilizado: material, forma, tamanho ou peso, como as configurações físicas de rede obrigatórias.

Interface

especifica ou restringe as funcionalidades inerentes a interface do sistema com usuário

Realizar requisitos

Exemplo (Classificação de requisitos arquiteturais)

Suportabilidade "O produto deve ser localizado, i.e. suportar varias linguas humanas"

Desenho "Persistência será controlada por uma base de dados relacional"

Implementação "A base de dados será PostgreSQL"

Confiabilidade "O sistema rodará 7 dias por semana, 24 horas por dia"

Funcional "Um sistema online de ajuda é requerido"

Implementação "A lógica de presentação será escrita em Python"

O porquê

- Ajuda a discussão com clientes
- Permite perguntar as boas perguntas
- Ajuda a ver as relações entre vários requisitos

Porque usar FURPS+ ?

- Muitas especificações esquecem os requisitos não-funcionais
- Melhorar a discussão sobre requisitos/especificações

Dica geral

Especificações/requisitos são negociáveis. Produtos não.

Resumo

- Definição de especificações informais
- Características
- Vantagens
- Desvantagens
- Exemplos
- Exemplos de especificações para o jogo da senha
- Método FURPS