

**Nr      Întrebarea / Вопрос**

**1      Paradigmele elaborarii produsului program / Парадигмы разработки программных продуктов**

1.1      Какова разница между free-lancing - ом и работой в команде с т.зрения технологий и инструментов разработки прог.продуктов ?

**Care-i diferența din punct de vedere al tehnologiilor și instrumentelor utilizate dintre free-lancing și lucrul în echipă ?**

Free-lancing: nu foloseste Jira sau alte tehnologii asemenea

Team: se foloseste tehnologia Jira cu paradigmele Agile, Waterfall etc.

---

1.2      Какие парадигмы (методологии) разработки прог.продуктов вам известны ?

**Ce paradigme de dezvoltare a produsului program cunoașteți ? Prin ce ele sunt commune și prin ce se deosebesc una de alta ?**

Paradigme:

Waterfall, Spiral, Fountain, Agile, Rapid Application Development, Build & Fix Model, Rapid Prototyping Model.

---

1.3      Каковы фазы развития проекта в водопадной парадигме разработки прог.продукта ?

**Care sunt fazele de dezvoltare a unui produs program în paradigma dezvoltării 'pas cu pas' (waterfall) ? Ce acțiuni de bază conțin aceste faze ?**

Fazele de dezvoltare:

1. Analiza ceintelor
2. Plan
3. Design
4. Programarea
5. Testarea
6. Livarea

Este o paradigma secventiala

---

- 1.4      Каковы фазы развития проекта в инкрементальной парадигме (Rational Unified Process - RUP) разработки прог.продукта ?

**Care sunt fazele de dezvoltare a unui produs program în paradigma dezvoltării incrementale (Rational Unified Process - RUP) ? Ce acțiuni de bază sunt prevăzute ?**

1. Inceput
2. Elaborare
3. Constructie
4. Tranzitie

Actiunile de baza:

Echipa stabileste daca se merita de inceput proiectul si ce resurse vor fi necesare.

---

- 1.5      Каковы фазы развития проекта в парадигме 'agile' разработки прог.продукта ?

**Care sunt fazele de dezvoltare a unui produs program în paradigma dezvoltării 'agile' ?  
Ce acțiuni de bază conțin aceste faze ?**

1. Brainstorm (analiza cerintelor)
2. Design (crearea prototipurilor)
3. Dezvoltarea (iteratii, pareri)
4. Asigurarea calitatii (iteratii, pareri)
5. Implementarea (productul, suport tehnic)

Dezvoltarea software unde echipa lucrează împreună pentru a dezvolta cu succes produse de calitate

Metodologie iterativa, bazata pe timp, deschisa catre clienti.

---

1.6      Какие роли в проекте разработки прог.продукта вам известны ?

**Ce roluri din cadrul unui proiect de dezvoltare a produsului program cunoașteți ? Ce funcții de bază ele îndeplinesc ?**

Manager ( se asigura ca fiecare persoana are un task, conduce work flow-ul, etc )

Designer

DB Designer ( se asigura ca baza de data e cat mai segura )

UI Designer ( se asigura ca interfata softului e cat mai moderna si ami placuta)

Tester ( testeaza soft-ul pentru a gasi bug-uri / moduri de a face improve la soft)

Software Architect

App Designer

Deployment Manager ( git pull and shit)

---

1.7      Что общего и в чем разница между ролью и сотрудником проекта по разработке прог.продукта ?

**Ce este comun și ce-i diferit între un rol și un angajat al proiectului ? Cum este rezolvată diferența dintre rol și angajat ?**

Ambii participa in dezvoltarea proiectului insa angajatul este clasificat ca intern(persoana ce este invatata si testata). Diferenta este rezolvata prin trecerea unei perioade de testare pentru intern si angajatarea lui ulterioara cu un rol anumit.

---

1.8      Перечислите несколько ролей, относящиеся к проекту, которых можно назвать 'Stakeholder'.

**Care-i esența termenului 'Stakeholder' și care-i rolul lui în proiect ? Numiți careva persoane, aferente proiectului, care pot fi numite 'Stakeholder'**

Stackholder = este orice persoana care are un interes in rezultatul proiectului.

1. Proprietarul produsului (Product owner)
2. Dezvoltator (Developer)
3. Utilizatorul final (End user)

- 
- 1.9      Что есть общего и чем отличается необходимость (needs) от требования (requirement) к программному продукту ?

**Ce are în comun și prin ce diferă necesitatea (needs) de cerința (requirement) față de produsul program ?**

Necesitatea este ceea ce se așteaptă de la program, iar cerința este informația folosită pentru a găsi necesitatea. Ambele pun un anumit standard față de produsul program

---

- 1.10    Каковы основные источники требований к программному продукту ?

**Care sunt sursele de bază ale cerințelor către produsul program ? Aranjați-le din câteva puncte de vedere.**

- Accesibil - furnizați informațiile necesare și ușor de găsit
- Curat - complet și fără greșeli
- Specific - identifică ce, când, de cine, cu ce efort
- precise - echilibrate (0,5 + mh, md, mw)
- Exacte - precizie previzibilă, erori în estimarea sarcinilor mici

sunt acceptabile

---

- 2      **Gestionarea cerințelor către un produs program / Управление требованиями к программному продукту**

- 2.1    Как преобразовать необходимость (needs) в требование (requirement) к программному продукту ?

**Ce este necesar pentru ca o necesitate să se transforme într-o cerință către produs program ? Cum să efectuăm o astfel de transformare ?**

1. Traduceți nevoile părților interesate, așteptările, constrângerile și interfețele în cerințe documentate ale clienților.

2. Stabiliți și mențineți o prioritate a cerințelor funcționale ale clientului și a atributelor de calitate.

2.A avea prioritate cerințele clienților ajută la determinarea domeniului proiectului, iterației sau creșterii. Această prioritizare asigură că cerințele funcționale și de atribut de calitate critice pentru client și alte părți interesate sunt abordate rapid.

3. Definiți constrângerile pentru verificare și validare.

---

2.2    Какие типы требований к программному продукту вам известны ?

**Ce tipuri de cerințe către un produs program cunoașteți ? Ce ele au comun și prin ce ele diferă ?**

Cerințe software

- Funcțional (FR) - acțiunile pe care trebuie să le poată efectua un sistem
  - Non-funcțional (NFR) - nedescris de FR
  - Utilitatea
  - Siguranta
  - Performanță
  - Suportabilitate
- Etc.
- 

2.3    Какая роль в проекте управляет требованиями к программному продукту и с какой целью ?

**Care-i scopul managementului cerințelor către un produs program și în ce el constă ?  
Căru rol din proiect îi aparține această funcție ?**

<>NE EBU<>

---

2.4    Какими артефактами оперируем при управлении требованиями к программному продукту ?

**Cu ce artefacte se lucrează la managementului cu cerințele către un produs program ?  
Care-i scopul fiecărui artefact ?**

ARTIFICATE DE INTRARE(input)

- Structura „echilibrată” a cerințelor pentru aplicație
- Un set de modele de aplicații
- Clase de aplicații, componente, subsisteme

ARTIFACTE DE IEȘIRE(output)

- O listă „echilibrată” și structurată de întrebări frecvente, cântărită cu ajutorul acesteia  
atribute (dependențe, priorități, efort etc.)
- Un set de diagrame de proiectare a produselor (proces, caz de utilizare etc.)

---

2.5      Какие атрибуты следует присвоить некоему набору требований к программному  
            продукту ?

**Ce atribute ar fi necesare de alocat unei cerințe față de un produs program ? Motivați  
răspunsul.**

RequirementId   Link

Risc

Effort

Nume

Descriere

Autor

Timestamp

Status

---

2.6      Что такое Work Breakdown Structure (WBS) при формировании дерева требований к  
            программному продукту ?

**Ce reprezintă Work Brakedown Structure (WBS) și cum ea poate fi utilizată la definirea arborelui cerințelor către produsul program ?**

WBS este folosit pentru a diviza lucrul in sub taskuri, facand lucrul mai usor si abordabil.  
Cu ajutorul ei putem diviza lucrul in mai multe branch-uri. Ex (Design, BD, Backend, Front)

-----

- 2.7 Что представляет собой сбалансированное дерево требований (ДТ) к программному продукту ?

**Ce reprezintă un arbore de cerințe ‘balansat’ și cum să-l obținem ?**

<>NE EBU<>

-----

- 2.8 В чем суть дилеммы ‘Commercial-of-the-shelf’ (COTS) или ‘Bespoke Application’ при разработке прог.продукта ?

**Care-i esența dilemei – ‘Commercial-of-the-shelf’ (COTS) sau ‘Bespoke Application’ la elaborarea unui produs program ?**

COTS - Solutia ( final product ) adaptata pentru a satisface cerintele cumparatorului.

Ceva facut pentru toti, industrial made / off the shelf. Este mai ieftin, dar este palid in comparatie cu Bespoke App.

Bespoke - Solutia custom care poate avea ceva mai in plus. Este folosita pentru a indeplini cerintele individuale, ceva facut custom doar pentru 1 pers

-----

- 2.9 Каковы выходные артефакты при разработке (implementation) прог.продукта ?

**Care sunt artefactele de intrare și cele de ieșire la elaborearea unui produs program ?**

De intrare:

- Structura „echilibrată” a cerințelor pentru aplicație
- Un set de modele de aplicații
- Clase de aplicații, componente, subsisteme

De iesire:

- O listă „echilibrată” și structurată de întrebări frecvente, cântărită cu ajutorul acestora atribute (dependențe, priorități, efort etc.)
- Un set de diagrame de proiectare a produselor (proces, caz de utilizare etc.)

---

2.10 Какие подходы к процессу кодирования прог.продукта вам известны ?

**Ce abordări ale procesului de implementare a unui produs program cunoașteți ? Care sunt avantajele fiecărei abordări ? Cum să alegem abordarea optimală de implementare produsului ?**

1. Construction
2. Compilation
3. Troubleshooting
4. Disassembling
5. Documenting
6. Unit Testing
7. Performance analysis
8. Collaboration
9. Etc.