|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тип | Наименование | прост | междинен | сложен | Общо |
| IT | Външен входен | x2 | x1 | х0 | 3 |
| OT | Външен изходен | х1 | х5 | х0 | 6 |
| FT | Вътрешен логически файлов | х2 | х2 | х1 | 5 |
| EI | Външен интерфейсен файлов | х0 | х0 | х0 | 0 |
| QT | Външен справочен | х0 | х5 | х1 | 6 |
| FC | Общо ненастроени функционални точки |  |  |  | 20 |

Външен входен тип -

* Прост - Справка на записани часове на пациент, Справка на записани часове на лекар - не е нужден вход в системата, съдържа малък брой типове данни, които само се визуализират
* Междинен - Вход в системата (използва се от разнообразни потребители) - не е прост, защото съдържа и се отразява на немалък брой типове данни, но броят им не е толкова голям

Външен изходен тип -

* Прост - Инициализира съобщение за грешни данни (или връща съобщение за успешна операция)- понеже съдържа една или две колони с грешни данни (за потребителско име/имейл и парола) и данните не се преобразуват при извеждането им
* Междинен - Управление на пациент, Управление на лекар, Управление на часове на леакар, Справка за записани часове на пациент, Справка за заети часове на лекар - отчетът съдържа повече от две колони, но не съдържа сложни операции и преобразования

Вътрешен логически файлов тип -

* Прост - Управление на пациент, Управление на лекар - малко типове записи, малък брой типове елементи от данни, не се изискват усилия за постигане на ефективност
* Междинен - Справка за заети часове на даден лекар, Справка за записани часове на даден пациент - типът не може еднозначно да се определи като прост или сложен, има различни типове данни и елементи
* Сложен - Управление на часове на лекар - понеже се изискват специални грижи за постигането на ефективност при записването и отменянето на час

Външен интерфейсен файлов тип - не виждам къде в системата може да се включи файл, който да се предава или използва и от друго приложение

Вътрешен логически файлов тип - сложността на външния справочен тип е равна на по-голямата от така определените две сложности (външен изходен тип и външен входен тип)

Вход в системата - външен изходен тип (междинен), външен входен тип (прост) -> междинен

Управление на пациент - външен изходен тип (междинен), външен входен тип (междиенен) -> междинен

Управление на лекар- външен изходен тип (междинен), външен входен тип (междиенен) -> междинен

Управление на часове на лекар - външен изходен тип (сложен), външен входен тип (междиенен) -> междинен

Справка на записани часове на пациент - външен изходен тип (прост), външен входен тип (междиенен) -> междинен

Справка на записани часове на лекар - външен изходен тип (прост), външен входен тип (междиенен) -> междинен

* Междинен - Вход в системата, Управление на пациент, Управление на лекар, Справка на записани часове на пациент, Справка на записани часове на лекар
* Сложен - Управление на часове на лекар,

1. Данните и управляващата информация се изпращат или получават по комуникационни линии - (не е споменато като изискване) - 0

2. Има разпределена обработка на данни - (обработват се взависимост от типа данни) - 2

3. Важно е достигането на висока ефективност - (изискване на клиента) - 5

4. Експлоатация върху силно натоварена операционна конфигурация – хардуер, софтуер - (не е опоменато от клиента, но е по-добре за производителността) - 2

5. Интензивността на транзакциите е висока - (транцакциите са от голяма важност особено при записването на час за лекар, не трябва да се случи няколко човека да се запишат за един и същи час, от което следва, че интензивността на транзакциите трябва да е висока) - 4

6. Наличен е интерактивен режим на въвеждане на данните - (по-лесена за ползване от хора без опит) - 4

7. Цели се ефективност от гледище на потребителя (изискване на клиента) - 5

8. Наличен е интерактивен режим на актуализирането на данните - (не е споменато като изискване, но би трябвало да информира потребителя за случващите се процеси при аклтуализация/редактиране на данните) - 2

9. Логиката на обработките е сложна - (не е споменато като изискване) - 0

10. Програмният код трябва да е reusable - (не е опоменато от клиента, но е добра практика) - 3

11. Цели се лесно инсталиране - (изискване на клиента) - 5

12. Цели се лесна експлоатация - (изискване на клиента) - 5

13. Може да се използва от разнообразни потребители - (изискване на клиента) - 5

14. Приложението е гъвкаво и лесно се модифицира - (не е опоменато от клиента, но е добра практика) - 2

PC (processing complexity) = 44

PCA = 0.65 + (0.01 x PC)

PCA = 0.65 + (0.01 X 44) = 1.09

PCA (processing complexity adjustment) = 1.09

FC (брой ненастроени функционални точки) = 20

FP = FC x PCA

FP = 20 x 1.09 = 21.8

FP (функционални точки) = 21.8

Effort = 0.6 \*FP + 0.001\*FP \* FP = 13.08 + 0.47524 = 13.55524

При наличието на достатъчно статистически данни не е трудно да се определи цената на създаване на една функционална точка

