Тема: Оценка качества и метрология программных средств. Для заданного преподавателем варианта разработать метрику качества, определить критерии обоснованности и свойства разработанной метрики.

Задание: Сделать модель качества ПС (набор характеристик и метрик продукта)

Для проверки веб-сайта были выбраны следующие характеристики, подхарактеристики, метрики:

1. Функциональная пригодность (w = 10)
   1. Полнота функций (w = 10)
      1. Запланированное кол-во методов API (Exp), реальное кол-во методов(Rea). Результат – Rea/Exp;
      2. Запланированное количество страниц (Exp), реальное кол-во страниц (Rea). Результат – Rea/Exp;
      3. Количество поддерживаемых платежных систем (Exp), реальное кол-во платежных систем (Rea). Результат – Rea/Exp;
      4. Количество полей, которые можно изменить в профиле (Exp), реальное кол-во полей, которые можно изменить (Rea). Результат – Rea/Exp.
   2. Защищенность (w = 1)
      1. Количество способов авторизации в системе (Exp), реальное кол-во способов авторизации (Rea). Результат – Rea/Exp;
      2. Использование SSL (Exp) - 1, реально (Rea) – 0. Результат – Rea/Exp;
2. Надежность(w = 9)
   1. Безотказность (w = 10)
      1. Среднее количество серверных ошибок (Ee) , реально (Er). Результат – 1 / (Er>Ee ? Er/Ee : 1)
      2. Количество клиентских ошибок, реально (Er). Результат – 1 / (Er>Ee ? Er/Ee : 1)
   2. Отказоустойчивость(w = 9)
      1. Процент успешно выполненных операций при возникновении серверной ошибки (Exp), реально (Rea). Результат – Rea/Exp
      2. Процент успешно выполненных операций при возникновении клиентской ошибки (Exp), реально (Rea). Результат – Rea/Exp
   3. Восстанавливаемость
      1. Количество успешных перезапусков сервера
3. Удобство(w = 7)
   1. Понимаемость(w = 10)
      1. Количество обращений к странице справки C к количеству страниц в приложении (P). Результат - 1 / (C / P === 0 ? 1 : C/P)
      2. Количество обращений в техподдержку (H), количество пользователей (U) - 100. Результат - 1 / (H/U === 0 ? 1 : H/U)
   2. Привлекательность(w = 8)
      1. Среднее время в приложениии в день, минут (Te), реально (Tr). Результат - Tr/Te
4. Сопровождаемость(w = 4)
   1. Анализируемость(w = 3)
      1. Количество времени разработчика, потраченного перед началом выполнения задания, часов (He), реально (Hr). Результат - 1/ (Hr/He === 0 ? 1 : Hr/He)
   2. Изменяемость(w = 5)
      1. Количество времени на внедрение нового функционала к среднему количеству времени выполнения задания (FTe), реально (FTr) . Результат - 1 / (FTr/FTe <= 1 ? 1 : FTr/FTe)
   3. Стабильность(w = 10)
      1. Количество багов B на количество тикетов (T). Результат - 1 / (B/T > 0.1 ? B/T : 1)
   4. Тестируемость(w = 5)
      1. Процент покрытия кода тестами (TCe, реально (TCr), Результат – TCr/TCe
5. Уровень производительности(w = 10)
   1. Временные характеристики(w = 10)
      1. Время загрузки страницы (Exp), реальное время загрузки (Rea). Результат – Rea/Exp;
      2. Время отклика базы данных (Exp), реальное время отклика (Rea). Результат – Rea/Exp;
      3. Время рендеринга страницы (Exp), реальное время рендеринга (Rea). Результат – Rea/Exp;
6. Переносимость(w = 10)
   1. Адаптируемость(w = 10)
      1. Количество поддерживаемых ОС (Exp), реальное количество поддерживаемых ОС (Rea). Результат – Rea/Exp;
      2. Количество поддерживаемых баз данных (Exp), реальное количество поддерживаемых баз данных (Rea). Результат – Rea/Exp;

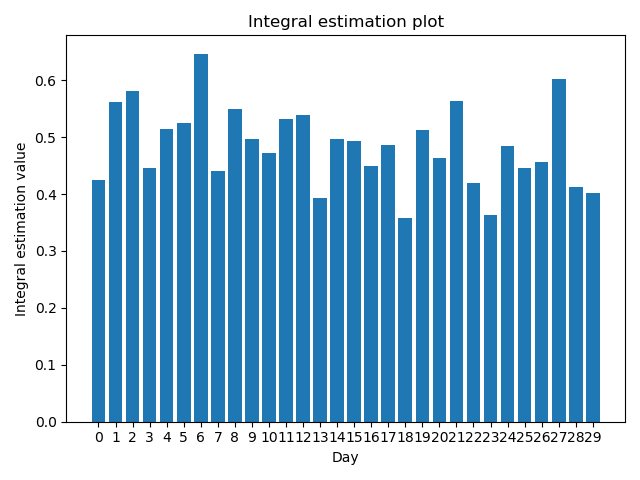


Рисунок 1. Гистограмма интегральных оценок по дням

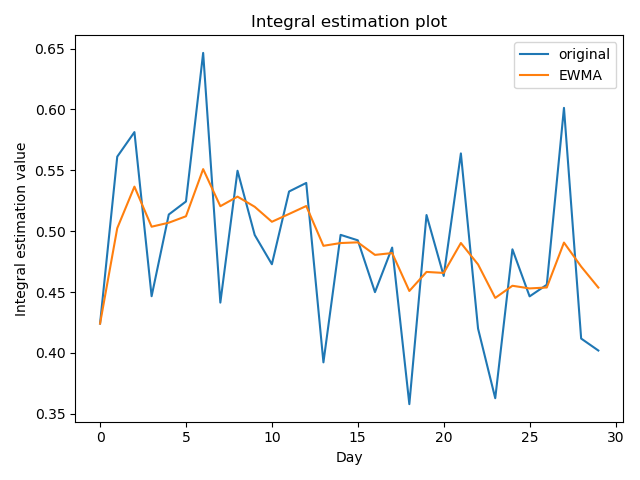


Рисунок 2. График инегральных оценок

Формат json-файла для хранения метрик:

{  
 **"quality\_model"**: {  
 **"characteristics"**: [  
 {  
 **"name"**: **"Функциональная пригодность"**,  
 **"importance-factor"**: 0.2,  
 **"sub\_characteristics"**: [  
 {  
 **"name"**: **"Полнота функций"**,  
 **"importance-factor"**: 0.7,  
 **"metrics"**: [  
 {  
 **"name"**: **"Запланированное количество методов"**,  
 **"importance-factor"**: 0.5,  
 **"values"**: [  
 0.1,  
 0.2,  
 0.3,  
 0.4,  
 0.5,  
 0.6,  
 0.7  
 ]  
 }  
 ]  
 }  
 ]  
 }  
 ]  
 }  
}

Скрипт для расчета интегральной оценки:

**def** calculate\_integral\_estimation(quality\_model, day):  
 characteristic\_estimation = 0  
 **for** characteristic **in** quality\_model[**"quality\_model"**][**"characteristics"**]:  
 sub\_characteristic\_estimation = 0  
 **for** sub\_characteristic **in** characteristic[**"sub\_characteristics"**]:  
 metric\_estimation = 0  
 **for** metric **in** sub\_characteristic[**"metrics"**]:  
 metric\_estimation += metric[**"importance-factor"**] \* metric[**"values"**][day]  
 sub\_characteristic\_estimation += sub\_characteristic[**"importance-factor"**] \* metric\_estimation  
 characteristic\_estimation += sub\_characteristic\_estimation \* characteristic[**"importance-factor"**]  
 **return** characteristic\_estimation  
  
  
**with** open(**"metrics.json"**, encoding=**"utf-8"**) **as** metrics\_f:  
 quality\_model = json.load(metrics\_f)  
  
**for** day **in** range(day\_quantity):  
 estimation.append(calculate\_integral\_estimation(quality\_model, day))

Скрипт для построения графиков:

y\_pos = np.arange(len(estimation))  
  
plt.bar(y\_pos, estimation, align=**'center'**)  
plt.xticks(y\_pos, list(range(day\_quantity)))  
plt.ylabel(**'Integral estimation value'**)  
plt.xlabel(**'Day'**)  
plt.title(**'Integral estimation plot'**)  
df = pd.DataFrame(estimation)  
ewma = df.ewm(span=7).mean()  
plt.figure()  
plt.plot(list(range(day\_quantity)), estimation, label=**"original"**)  
plt.plot(list(range(day\_quantity)), ewma, label=**"EWMA"**)  
plt.ylabel(**'Integral estimation value'**)  
plt.xlabel(**'Day'**)  
plt.title(**'Integral estimation plot'**)  
plt.legend()  
plt.show()

Для автоматизации процесса расчета модели качества было разработано спец. программное обеспечение. На основании результатов работы ПО была получена интегральная оценка качества программного средства «Сервис P2P-кредитования».