

АНАЛИЗ КАЧЕСТВА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ИСПОЛЬЗОВАНИЕМ КОНТРОЛЬНЫХ СПИСКОВ

Матвеев В.А.

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

*Образцова О.Н. – и.о. зав. кафедрой ИСиТ, к.т.н., доцент
Бакунова О.М. - иссл.т.н., ст преподаватель*

В статье раскрывается понятие контрольный список, его назначение, перечислены виды списков, приведен пример.

Контрольный список (перечень, чек-лист; англ. checklist) — список утверждений, в котором каждое утверждение может быть либо подтверждено, либо опровергнуто. При этом, пройденным считается тот контрольный список, где были даны желаемые ответы. В качестве примера можно привести список дел (to-do list), где каждое из них должно быть выполнено.

Обеспечение качества при использовании контрольного списка означает проверку степени соответствия компонентов системы перечисленным в списке требованиям. Сложность составления перечня зависит от разрабатываемой системы и необходимого уровня детализации.

Контрольный список может быть составлен таким образом, чтобы его можно было использовать повторно для различных разрабатываемых систем, и он может не включать в себя строго регламентированную последовательность шагов. Таким образом, контрольный список является гибким инструментом при анализе качества любых систем, в том числе и систем обработки информации.

Контрольные списки в области качества разработки программного обеспечения можно разделить на множество категорий:

- 1) анализ проектируемой системы;
- 2) конструирование и проектирование;
- 3) компоненты кода: классы, методы и прочие конструкции;
- 4) данные и их типы;
- 5) операторы;
- 6) тестирование и отладка;
- 7) совместное конструирование;
- 8) прочие контрольные списки.

Пример контрольного списка – Качество требований [1]

<input type="checkbox"/>	Написаны ли требования на языке, понятном пользователям? Согласны ли с этим пользователи?
<input type="checkbox"/>	Нет ли конфликтов между требованиями?
<input type="checkbox"/>	Определено ли приемлемое равновесие между параметрами-антагонистами, такими как устойчивость к нарушению исходных предпосылок и корректность?
<input type="checkbox"/>	Не присутствуют ли в требованиях элементы проектирования?
<input type="checkbox"/>	Согласован ли уровень детальности требований? Следует ли какое-нибудь требование определить подробнее? Менее подробно?
<input type="checkbox"/>	Достаточно ли ясны и понятны требования, чтобы из можно было передать независимой группе конструирования? Согласны ли с этим разработчики?
<input type="checkbox"/>	Каждое ли требование релевантно для проблемы и её решения? Можно ли проследить каждое требование до его источника в проблемной среде?
<input type="checkbox"/>	Можно ли протестировать каждое требование? Можно ли будет провести независимое тестирование, которое позволит сказать, выполнены ли все требования?
<input type="checkbox"/>	Определены ли все возможные изменения требований и вероятность каждого изменения?

Список использованных источников:

1. Макконнелл С., Совершенный код. Мастер-класс / Пер. с англ. – М. : Издательство «Русская редакция», 2014. – 896

стр.