

**Климович Татьяна Михайловна**

магистрант

**Серебрякова Татьяна Александровна**

канд. экон. наук, доцент

ФГБОУ ВО «Тихоокеанский государственный университет»

г. Хабаровск, Хабаровский край

## **ПРОГРАММНАЯ ИНЖЕНЕРИЯ**

*Аннотация: в статье рассматриваются основные проблемы развития программной инженерии, чем вызвано его становление и развитие и то как, он влияет на современную экономику страны.*

**Ключевые слова:** программа инженерия, программирование, экономика.

Программная инженерия как область знаний зародилась относительно недавно – в середине 60-х годов прошлого века. За сорок лет своего существования программная инженерия прошла достаточно бурный путь развития в современной российской и мировой экономики. Не будет большим преувеличением сказать, что путь программной инженерии это путь поиска путей выхода из кризисов, периодически возникающих за все время разработки программ для компьютеров. Эти кризисы приводят к большим материальным затратам, так как приходится улучшать ПО программ. На этом пути накоплен и обобщен громадный опыт как успешных проектов, так и неудач, и в настоящее время вряд ли возможно найти специалистов, занятых в экономике не использующих удачные методы и приемы программной инженерии [1].

Актуальность программной инженерии заключается в том, что за последние несколько десятилетий она превратилась в экономическую дисциплину со множеством направлений: тестирование, программирование, проектирование, архитектура, поддержка, измерения и эволюция. Появляются все новые публикации и конференции, однако о них быстро забывают, а программная инженерия продолжает самостийно развиваться вне традиций – в какой-то мере это и ремесло, и искусство, и логика [2].

Существует множество проблем в программной инженерии, так как ни у кого нет волшебного рецепта, как создать надежное программное обеспечение, способное успешно работать в любых экономических проектах и организациях, – имеется множество тонких вопросов, которые, возможно, никогда не получат универсального ответа. Но мы рассмотрим несколько основных проблем и попробуем решить их.

### *Проблема 1: совместимости*

Для решения этой проблемы мы должны обратиться к качеству ПО. Для того чтобы скомпоновать два программных модуля, важно учесть вероятность непредусмотренного поведения.

При объединении различных компонентов в одну систему необходимо, по крайней мере, для каждого из них в отдельности, учитывать:

- уникальность версии программного обеспечения;
- уникальность аппаратного обеспечения, на котором устанавливается компонент;
- операционную среду;
- пространство внутренней и внешней угрозы;
- поддерживаемые политики и процедуры;
- параметры качества, которые были разработаны для описания поведения;
- время.

Если попытаться связать эти семь элементов одного компонента с элементами другого компонента (а они могут различаться), вам будет трудно предсказать, как компоненты поведут себя при объединении.

Кроме того, обратите внимание на роль, которую интерфейсы играют в этом объединении. Они могут оказаться столь же ненадежными и так же содержать ошибки, как и объединяемые компоненты. Мы много лет ждали, но придется подождать еще, пока не появится продуманная, масштабируемая теория, позволяющая разрешать неоднозначность таких взаимосвязей. Не хотелось бы возвращаться к временам «спагетти»-логики в интерфейсах.

Что касается моего решения данной проблемы, то я бы дала совет: проектировать максимально возможный уровень адаптируемости в компонентах. Существуют аспекты, в большей степени определяющиеся внешним миром, чем основной функциональностью, которые позволяют так реализовывать объединение кода компонента с другими, чтобы можно было повторно его использовать с минимальными доработками.

*Проблема 2:* высокая стоимость ПО связана с повторной разработкой одинаковых фрагментов программ.

Решение: модульное программирование – это организация программы как совокупности небольших независимых блоков, называемых модулями, структура и поведение которых подчиняются определенным правилам. Это ключевое понятие ввел ученый Дэвид Парнас в 1972 году, чтобы помочь программистамправляться со все более и более сложными программными системами.

Использование модульного программирования позволяет упростить тестирование программы и обнаружение ошибок, упрощается процесс повышения эффективности программ, так как критичные по времени модули могут быть написаны на языках низкого уровня (ассемблерах) и многократно переделываться независимо от других. Кроме того, модульные программы могут разрабатываться параллельно во времени многими исполнителями, что сокращает сроки их разработки и испытаний, их значительно легче понимать, а модули могут использоваться как строительные блоки в других программах.

С моей точки зрения, перечисленные проблемы самые сложные и требуют неотложного внимания со стороны исследователей. Они открывают огромные возможности и определяют план будущих исследований для сообщества программной инженерии. Многие уже работают в этих областях, но реальные результаты появляются крайне медленно. Эти проблемы в большей степени относятся именно к программной инженерии в экономике.

Сегодня индустрия программирования – одна из наиболее перспективных и динамичных отраслей мировой и современной российской экономики, в которой представлены основы экономики производства сложных программных

продуктов высокого качества, которые базируются на традиционных принципах и методах экономики разработки сложных технических систем. Создание таких программных продуктов связанных с большими затратами, определяет необходимость их строгого планирования, формализации и стандартизации производственных процессов, а также экономического контроля и сопровождения, подобных применяемых при производстве других крупных промышленных изделий. Таким образом, при создании ПО нужно учитывать актуальность продукта, чтобы не пришлось вносить изменения в ПО, которое приведет к внеплановым финансовым затратам и потери времени.

### ***Список литературы***

1. Карпенко С.Н. Введение в программную инженерию.
2. [Электронный ресурс]. – Режим доступа: [http://www.infosystem.ru/software\\_engineering/software\\_engineering\\_13\\_problem\\_se.html](http://www.infosystem.ru/software_engineering/software_engineering_13_problem_se.html)
3. [Электронный ресурс]. – Режим доступа: <http://faculty6.khai.edu/site/istoriya-kafedri-1.html>
4. Чертова дюжина проблем программной инженерии [Электронный ресурс]. – Режим доступа: <https://www.osp.ru/os/2007/07/4391815/> (дата обращения: 12.02.2018).