

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет

Кафедра

ЭССЕ

Системная инженерия на примере системы «Компьютерная игра»

Выполнил: студент ? курса магистратуры

.....

Воронеж 2016

Современные компьютерные игры являются одним из самых сложных видов программного обеспечения, требующего большого числа человеческих и иных ресурсов, а также использования новейших технологических решений и алгоритмов в области графики, искусственного интеллекта, сетевого взаимодействия и т.д.

Т.к. любая система может рассматриваться с разных точек зрения, оговорим заранее, что в качестве целевой системы рассматривается сама компьютерная игра как программный продукт.

Основные функции системного инженера в игровой индустрии как правило выполняет ведущий геймдизайнер. Именно он отвечает за проработку проекта в целом.

Всякая система включает две составляющие (**единство**): её функцию и конструкцию.

Основные *функции* компьютерных игр: психологический отдых и эмоциональная разрядка, развитие памяти, внимания, рефлексов. Кроме того, компьютерная игра может являться произведением искусства, а, следовательно, формировать некий художественный образ и выполнять функцию доставления эстетического наслаждения игроку.

Функция системы обеспечивается её *конструкцией*. Среди конструктивных частей любой компьютерной игры можно выделить графический «движок», физический «движок», подсистему искусственного интеллекта, интерфейс меню, интерфейс игрового экрана, игровые уровни, звуковую подсистему и т.д.

Связь функции и конструкции системы задаётся её *архитектурой*. Для любой компьютерной игры создаётся её архитектурное описание в виде набора артефактов:

- Концепт-документ. Содержит описание концепции и основных идей.

- Дизайн-документ (диздок). Содержит описание сюжета, игрового мира и персонажей, геймплея (игрового процесса), интерфейса, управления и т.д.
- Диаграммы и схемы, отражающие основной компонентный состав ПО и протоколы связи между компонентами.

Таким образом, концепт-документ и диздок предназначены для подробного описания того, как система должна реализовывать свою основную функцию, а специальные диаграммы отражают компонентный состав системы, необходимый для осуществления этих задумок. Вместе такой набор документов и обеспечивает описание связи функции и конструкции.

Требования к системе формируются стейкхолдерами (заинтересованными лицами). Перечислим основных стейкхолдеров:

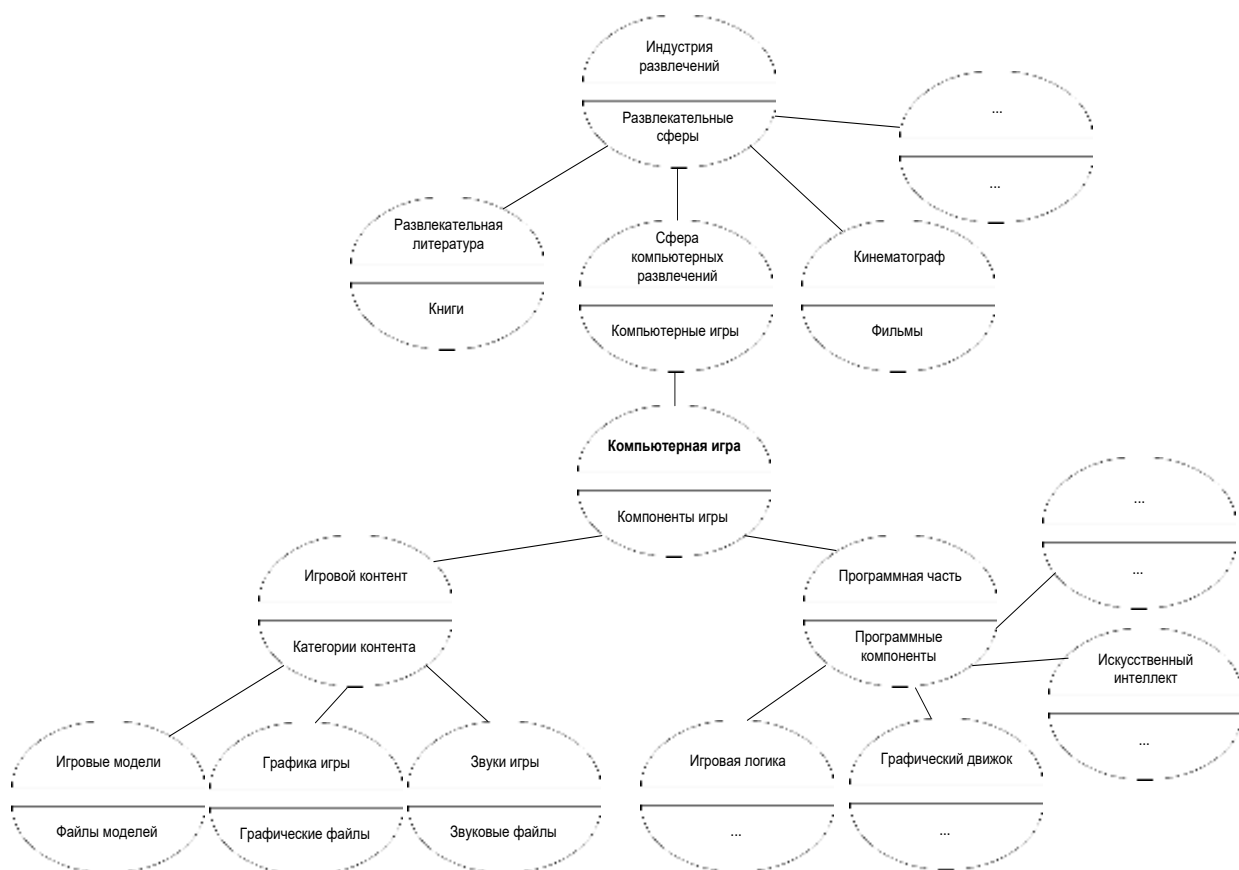
- *Разработчики* заинтересованы в создании стабильного программного продукта (требующего минимальную дальнейшую поддержку) в кратчайший срок, а также в максимизации продаж.
- *Геймдизайнер(-ы)* является творческим координатором и автором идей, потому заинтересован в максимально полной и точной реализации своих задумок.
- *Дистрибьютор* заинтересован в максимизации числа проданных копий игры и итоговой прибыли.
- *Игроки (геймеры)* также предъявляют ряд требований, формулировки которых зависят от текущей моды, предыдущих проектов студии-разработчика и многих других факторов.
- *Родители* несовершеннолетних игроков заинтересованы в минимизации насилия, нецензурной лексики и эротического содержания в играх, наличии поучительных и развивающих элементов.

- *Рейтинговая комиссия.* Предъявляет список требований, которым должна удовлетворять игра, чтобы получить тот или иной возрастной рейтинг.

Ограничения формируются системным инженером. Они могут касаться:

- Используемых технологий, библиотек, программных средств
- Допустимости внедрения «взрослого» контента
- Системных требований
- Производственного бюджета
- и т.д.

Как и всякая система, компьютерная игра может рассматриваться как *холон* и входить в ту или иную *холархию* (иерархию холонов). Приведём пример возможной холархии, изобразив её в виде диаграммы «гамбургера»:



Как известно, помимо целевой различают также обеспечивающую систему и систему в эксплуатационной среде.

В качестве *обеспечивающей системы* выступает штат сотрудников фирмы-разработчика и фирмы-дистрибьютора, компьютеры и другие аппаратные средства, с применением которых ведётся разработка, используемое программное обеспечение, база имеющихся готовых наработок отдельных элементов игр (движки, графические элементы и прочее) и т.д.

К *системе в эксплуатационной (операционной) среде* можно отнести интернет, компьютеры и игровые приставки геймеров, электросеть, питающая оборудование офиса, тренды игровой индустрии и т.д. (игроки - ?)

Весь период существования системы принято делить на *стадии жизненного цикла*. На каждой такой стадии выполняется ряд практик. Выполнение одной и той же практики может растягиваться на несколько стадий жизненного цикла, лишь меняя свою интенсивность.

Обычно разработка компьютерных игр протекает итерационно: некоторый объём функциональности реализуется в рамках отдельной итерации (пройдя все стадии жизненного цикла), после чего происходит анализ полученной версии игры и переход к новой подобной итерации. Процесс останавливается, когда версия игры достигает необходимого уровня готовности.

Рассмотрим стадии жизненного цикла системы в соответствии с так называемой V-диаграммой.

- *Определение требований*

На данной стадии происходит сбор и анализ требований к будущей игре, которые формулируются стейкхолдерами.

Требования могут касаться:

- Целевого возрастного рейтинга
- Продолжительности прохождения
- Сложности игры
- Наличия сетевого режима
- Необходимости выпуска на других платформах (игровые консоли, другие операционные системы)

- Минимальных системных требований
- и т.д.

- *Архитектурное проектирование*

Продумывается концепция игры, геймплей, игровой мир, персонажи и т.д. Формируется представление о компонентном составе игры и способах взаимодействия отдельных компонентов. Создаётся архитектурное описание системы в виде набора артефактов: концепт-документа, диздока, диаграмм, отражающих архитектуру ПО.

- *Рабочее проектирование*

Различными специалистами происходит проектирование уже отдельных компонентов и подсистем игры. Программные компоненты обычно документируются диаграммами и текстовыми описаниями в соответствии со стандартом UML или любым другим. Дизайнерами и художниками создаются макеты интерфейса, дизайна персонажей, локаций и прочих графических элементов. Принимается решение об использовании готовых сторонних графических, физических движков или других компонентов.

- *Изготовление*

Отдельные компоненты игры (графический «движок», физический «движок», сетевая подсистема и др.) создаются или настраиваются под нужды проекта (если используются готовые). Программистами разрабатываются и внедряются программные элементы, художники и дизайнеры рисуют графический контент в соответствии с ранее созданными макетами, композиторами пишется музыкальное сопровождение, 3D-моделлеры заняты созданием моделей различных объектов, их анимаций, ландшафта. Проводится модульное тестирование программных компонентов.

- *Интеграция*

Компоненты игры объединяются воедино в соответствии с описанной ранее архитектурой. Производится интеграционное тестирование компонентов в связке.

- *Приёмка в эксплуатацию и эксплуатация*

После завершения очередной итерации и получения новой версии игры проводится окончательная верификация и валидация системы. В зависимости от степени готовности и результатов тестирования происходит либо начало новой итерации, либо остановка основной разработки. Во втором случае получают релизную версию игры, которую затем подготавливают (выполняется обфускация и другие меры защиты, оформляется инсталлятор) и тиражируют на физические и цифровые копии. Копии игры распространяются через соответствующие магазины или интернет, игра эксплуатируется конечными пользователями – игроками, купившими её.

В стандарте ISO 15288:2008 указаны 25 обязательных практик системной инженерии. Опишем технические практики, т.к. именно они непосредственно связаны с целевой системой.

- *Сбор требований*

Организуются мероприятия, нацеленные на выяснение требований заинтересованных сторон: опросы целевой аудитории, переговоры разработчиков и дистрибьюторов, собрания сотрудников.

- *Анализ требований*

Собранные сведения анализируются на предмет противоречивости. Организуются дополнительные мероприятия, нацеленные на уточнение требований и поиск компромиссных решений.

- *Архитектурный дизайн*

Геймдизайнером прорабатывается концепция игры, детали геймплея и игрового мира. Формируется описание необходимых компонентов и интерфейсов для их взаимодействия. Результатом выполнения практики является архитектурное описание в виде набора перечисленных ранее артефактов.

- *Изготовление*

Коллективами специалистов создаются соответствующие компоненты игры. Для программных элементов получают исходные коды и

скомпилированные библиотеки или исполняемые файлы. Мультимедиа контент создаётся в подходящем программном обеспечении и рендерится в готовые звуковые или графические файлы. Конструируются и рендерятся модели объектов и ландшафта.

- *Интеграция*

Модели объектов и окружения объединяются в игровые уровни. Готовый графический и звуковой контент начинает использоваться графическими и звуковыми программными компонентами, искусственный интеллект внедряется в код для управления игровыми персонажами и событиями и т.д.

- *Верификация*

Верификация производится главным образом при помощи различных автоматизированных и ручных тестов. Выполняется тестировщиками. Цель тестирования – поиск ошибок и недоработок в коде, интерфейсе и других компонентах. Может проводиться тестирование производительности, безопасности, юзабилити и т.д. Верификация также продолжает осуществляться в процессе альфа и бета-тестирования.

- *Переход к эксплуатации*

Релизная версия игры подготавливается и тиражируется на физических носителях и распределяется по игровым магазинам. Цифровая копия игры размещается в онлайн-магазинах. Если игра или её отдельные режимы требуют наличия сторонних выделенных серверов, то они арендуются у дата-центров или создаются на базе мощностей самой компании-разработчика.

- *Валидация*

Валидация осуществляется путём предоставления копии текущей версии игры некоторому кругу добровольцев, относящихся к целевой аудитории. Игроки эксплуатируют предоставленную копию и сообщают о своих впечатлениях, а также о найденных ошибках (багах), тем самым осуществляя и валидацию, и частично верификацию системы. Если игра в этот момент находится на ранней стадии и содержит только основные

функциональные возможности (так называемая альфа-версия), то такие мероприятия называются *альфа-тестированием*. Как правило, оно проводится внутри организации, без привлечения сторонних тестеров. Тестирование цельной, но ещё сырой и недоработанной версии называется *бета-тестированием*. Оно обычно осуществляется с привлечением сторонних тестеров. Если список участников теста строго ограничен, то он называется *закрытым*. Если же любой желающий может подать заявку на участие – *открытым*.

- *Эксплуатация*

Конечные пользователи приобретают копии игры в магазинах розничной продажи, через онлайн-сервис Steam или его аналоги и играют.

- *Обслуживание*

Продолжается пост-релизная доработка игры при помощи выпуска патчей, обновлений, внутриигрового контента. Обслуживаются игровые сервера или производится регулярная оплата аренды сторонних серверов, если они требуются. Служба поддержки работает с поступающими от игроков вопросами и сообщениями о проблемах.

- *Вывод из эксплуатации*

Хотя фактическая эксплуатация игры не прекращается до тех пор, пока копии игры используются конечными пользователями, в жизненном цикле любой компьютерной игры предусмотрены действия, направленные на окончание поддержки игры её авторами. Эти действия осуществляются при устаревании и пропадании спроса на игру или при ликвидации компании-правообладателя. В этом случае игра снимается с продажи, прекращается процесс разработки патчей, обновлений, игрового контента, если он продолжался до этого момента. Освобождаются арендованные сервера. Иногда игра выкладывается в бесплатный доступ или даже с открытым исходным кодом.