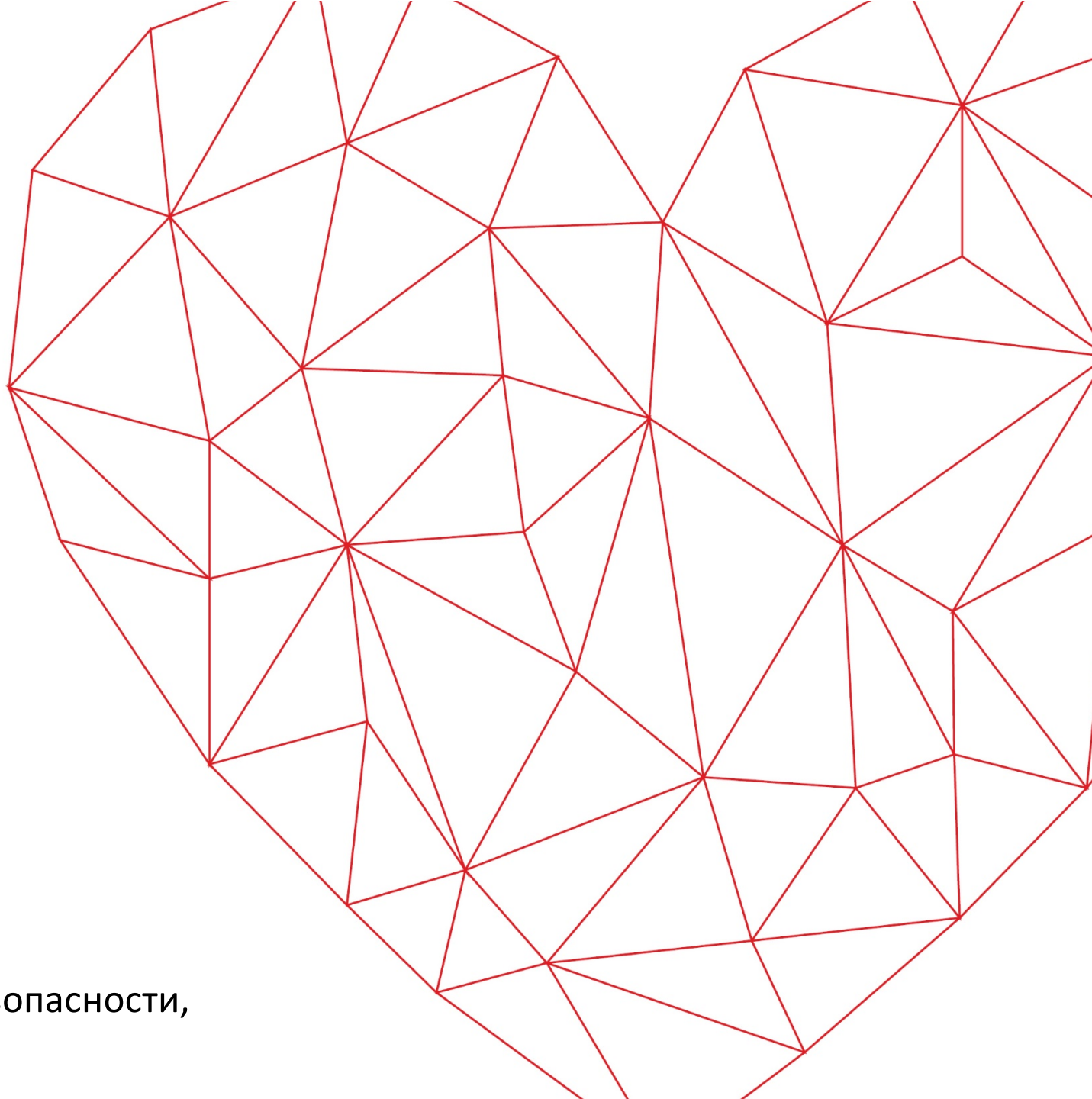
SUNLIGHT

Secure SDLC: спецификация классификации методологий безопасной разработки



Подготовил: Шмаков Илья, CISO, AppSec

Заместитель директора по информационной безопасности,

2021, материал для ITSec

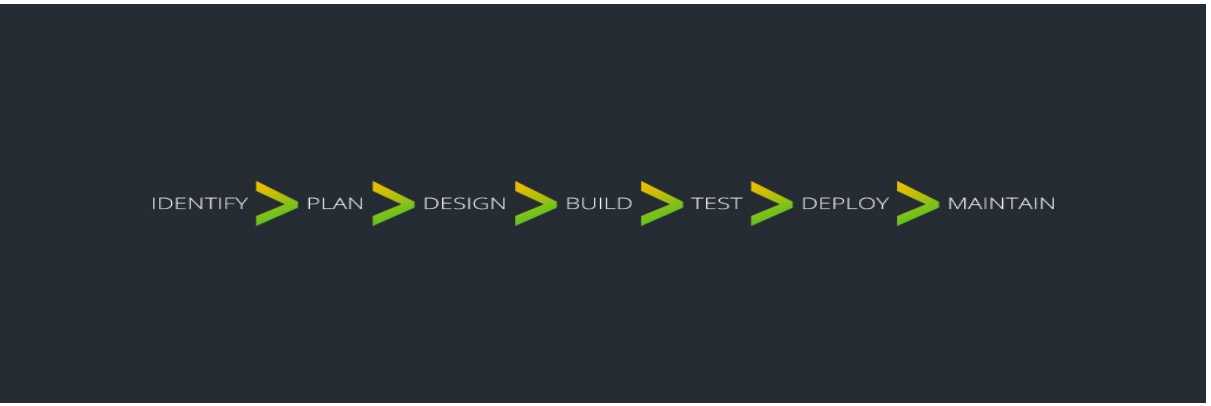
Аннотация

- Представляется спецификация методов, средств и методологий безопасной разработки, принятых на территории РФ, включая перенятый опыт от зарубежных коллег, "бестпрактик".
- Материл рассматривается со стороны коммерческого сектора и самых эффективных форматов безопасной разработки для бесперебойной работы бизнеса. Рассматривается в практической части специфик.
- Целью является передача опыта и аналитика данных применяемых методологий, методов и средств разработки в защищенном исполнении. Представлен анализ, соответствующий разбор действующего рынка, законодательства, в части касающейся разработки информационных систем и сред в защищенном исполнении, которая с каждым днем набирает "критическую массу" в различных отраслевых компаниях из-за ужесточения рекомендаций и требований регуляторов ИБ.
- Безопасная разработка критична из-за числа рисков, инцидентов ИБ – которые критически растут с каждым днем. Прогрессирующие злоумышленники стали понимать принципы разработки модели **win2win** организаций, которые подвергаются атакам, при этом им удается входить в группу доверенных пользователей из-за недостатка компетенций сотрудников.

SUNLIGHT

Принцип SDLC, в части касающейся Secure

- Это такой замкнутый цикл, в котором каждый этап влияет на действия в последующих и дает перспективные указания на будущее.
- Для получения ответов на конкретные вопросы и обеспечения согласованности вашего процесса разработки все шесть этапов стараются эффективно и последовательно друг на друга влиять.



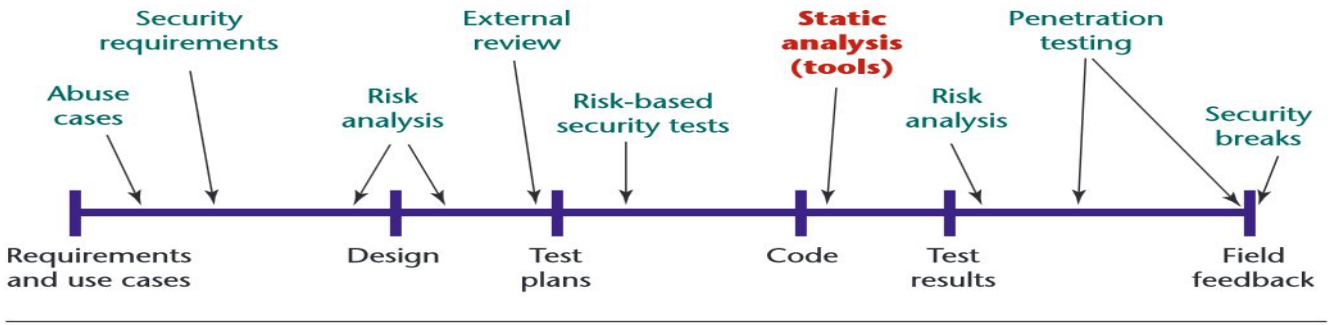


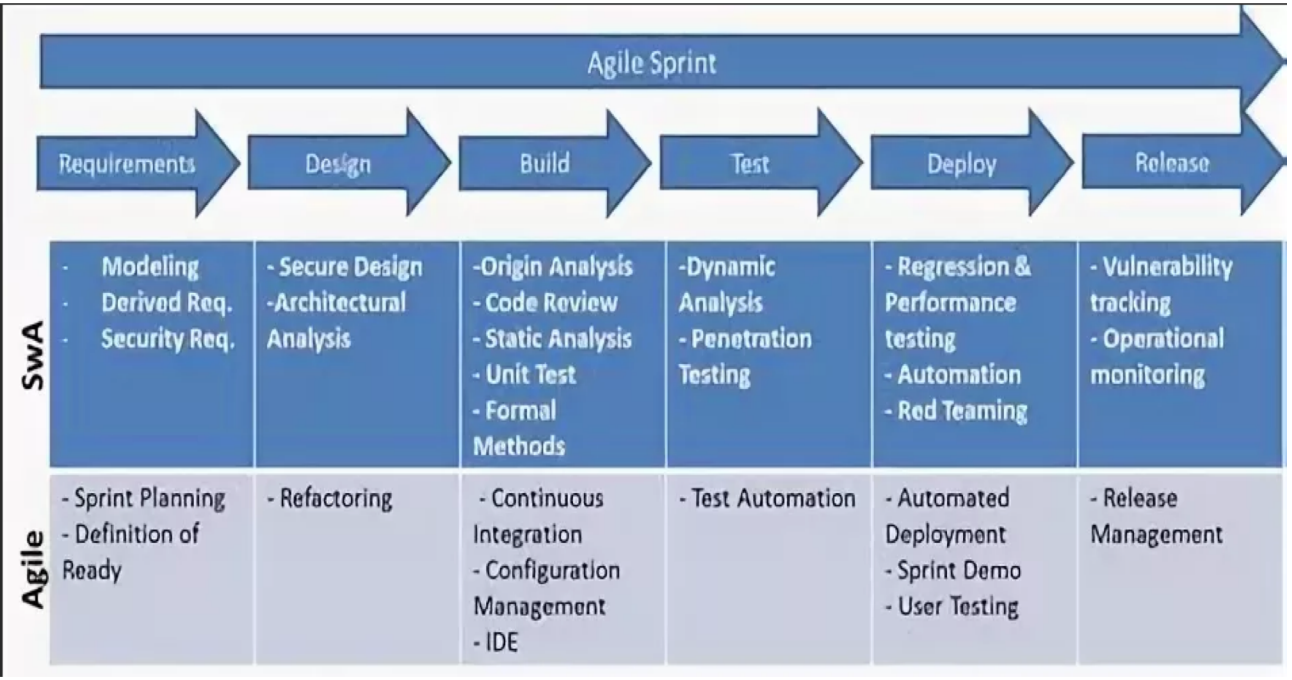
Figure 1. The software development life cycle. Throughout this series, we'll focus on specific parts of the cycle; here, we're examining static analysis.

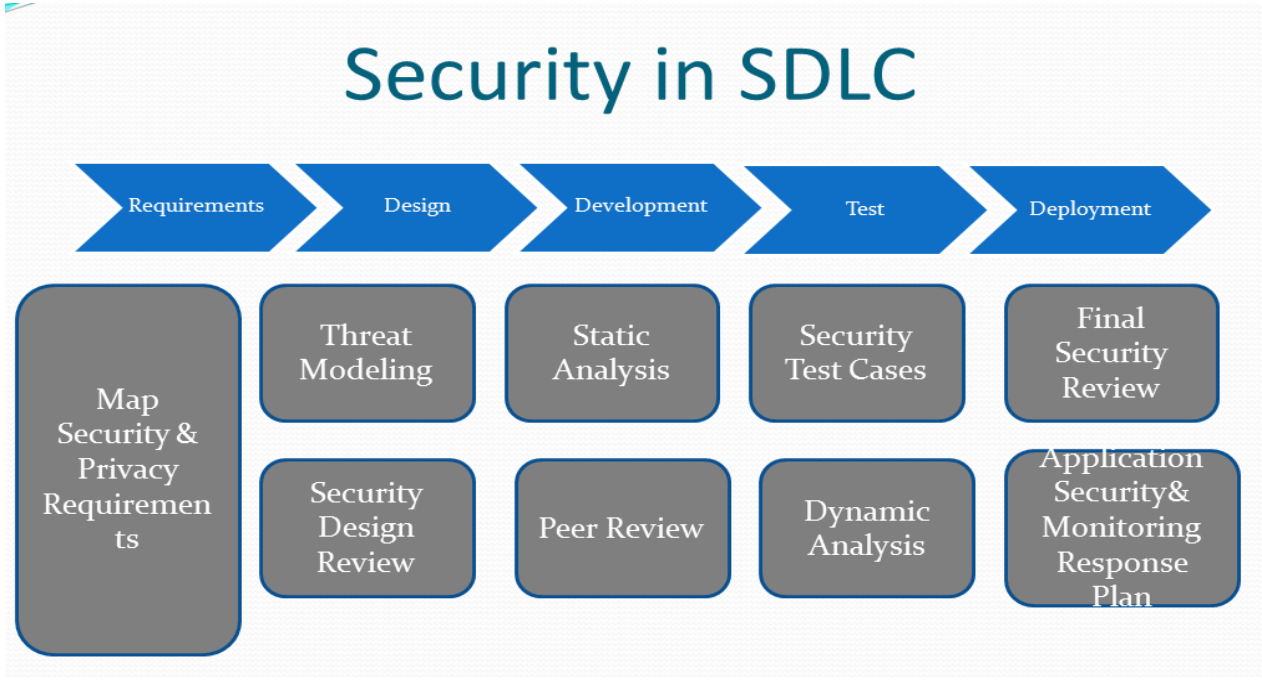
- **Анализ требований** отвечает на вопрос «Какие проблемы требуют решений?»
- **Планирование** отвечает на вопрос «Что мы хотим сделать?»
- **Проектирование и дизайн** отвечает на вопрос «Как мы добьемся наших целей?»
- **Разработка ПО** регулирует процесс создания продукта.
- **Тестирование** регулирует обеспечение качественной работы продукта.
- **Развертывание** регулирует использование финального продукта.

SUNLIGHT

Принципы Agile Sprint, в части касающейся Secure

- Наивысшая ценность — это удовлетворение потребностей заказчика благодаря регулярной и максимально ранней поставке ценного для него ПО и всегда готовы изменять требования.
- Бизнес обязательно должен работать вместе с программистами, помогать им понять специфику данного рынка.





- Наилучших результатов достигает команда замотивированных профессионалов.
- В Agile важен ритм, постоянные улучшения.
- Agile вообще не будет работать, если вы написали ***-код.
- Команда должна постоянно анализировать свою работу, процессы.

SUNLIGHT

Введение

- Материал базируется на принципах и процессах автоматизации ИС и сред организации, где рассматривается вопрос со стороны личной практики как руководителя, так и разработчика.
- В общепринятом представлении на рынке в потребительской сфере отсутствует понимание построения структуры и непосредственно семантики работы ИС и сред.
- Методы, средства и методологии разработки по созданию, оптимизации, масштабируемости, итерационной интеграции, введению в промышленную эксплуатацию – каноничны, во всем промежутке времени, которые парраллелелись по различным специфичным и обособленным между собою направлениям с применением разного рода подходов и методов для их реализаций. В последствии чего стали возникать угрозы, риски и последствия инцидентов для организаций, которые занимались обработкой, хранением информации, в том числе остро возник вопрос по ОИБ (обеспечение ИБ) информации: целостности, доступности, конфиденциальности.
- В последствии данных форматов и форм-факторов возник вопрос безопасной разработки в организациях, где направление стало развиваться в геометрической прогрессии и набирать экспоненциальные обороты.

Универсальные характеристики ИС в безопасной разработке

- Определим термин *безопасность информации*, который будем понимать как состояние уровня ее защищенности, которое обеспечивается сохранением качественных и количественных характеристик, а именно — целостности, доступности, конфиденциальности.
- Основное назначение функционирования: сбор, хранение и обработка информации;
- Нацеленность функционирования под потребительские нужны и цели, при необходимости реализации: объединения и распараллеливания функциональных возможностей на подразделения и структуры, непосредственно самой организации;
- В следствии сбора и обработки информационных данных, последующего воссоздания, воздействия и изменения информации — на ней отражается политика взаимодействия процессов анализа и синтеза, представляемая обобщением организационных, технологических, технических, программно-аппаратных и информационных средств, и методов;
- Представление проектирования интерфейса, в том числе соотношения **UI/UX**, прикладного программно-аппаратного обеспечения, как принцип минимальной содержательной достаточности для представление реализации взаимодействия функционирования с исключением эксплуатационной возможности НДВ, в целом и частном.

SUNLIGHT

Общая спецификация методологий по безопасной разработки ПО

- Под *спецификацией методологии* понимается — объединение перечня средств и методов, применяющихся в специфике деятельности организации, со стороны теории и практики разработки, включая прототипирование.
- *Методологией* является формат объединения методов, средств и технологий, применяющихся во время всего жизненного цикла процесса разработки и прототипирования ПО в организации.
- Стоит отметить, что также под методологией безопасной разработки ИС понимается организация процессов прототипирования и выстраивания ИС и сред, таких как: обеспечения управления процессами для гарантированного выполнения и реорганизации процессов.
- Целью методологии является присвоение рамок конечного функционирования и бесперебойной работоспособности.
- В реализации применяются специализированные подсистемы ИБ для формирования средств и методов разработки и прототипирования посредством политик ИБ организации в СУИБ.

SUNLIGHT

Задачи для обеспечения Secure SDLC

- Представления разработки и внедрения в промышленную эксплуатацию АИС, отвечающей целевым нуждам и спецификации по политикам работы организаций, в том числе ТТ и законодательных актов РФ, а также внутренних политик организаций и требований;
- Предоставления гарантий выполнения требований действующих регуляторов из специфики организации и приложений на разработку и интеграцию АИС с заданными параметрами, относительно ТТ и ТЗ в течение утвержденного календарного плана, а также оговоренного бюджета на разработку. Данный формат также может рассматриваться исходя из применяемых методов разработки ПО;
- Представления сопровождения технической и методологической части обучения, также модификации и масштабирования системы для обеспечения соответствия целевых нужд организаций, в том числе формата поддержания S.M.A.R.T.;
- Представления обеспечения разработки и внедрения в промышленную эксплуатацию с требованиями оптимизированности, переносимости, масштабируемости, адаптивности;
- Представления возможности вариативного использования в АИС разработанных средств и методов в программно-аппаратном обеспечении, СУБД, СУИБ и тому подобное. Отметим, что данный формат зависит от политик применяемых организацией, а также ее сферы деятельности.

SUNLIGHT

Фазы SDLC

1. Planning & Analysis

Gather all relevant information from stakeholders and analyze this information to determine what will be feasible.

2. Design

Create a design specification. This should answer the question, "How will we build this?".

3. Implement (or Code)

Building the software using a programming language by the development team.

4. Testing & Integration

Release the application to a testing environment, test all areas of the application to spot any defects, bugs, or other problems.

5. Deployment

All defects have been fixed and the software has been built to the agreed-upon goals and specifications.

6. Maintenance

Updating and supporting the software after it has been delivered to the market.

SUNLIGHT

Классификация Secure SDLC, Часть 1

1.1. Методология императивного программирования, которая характеризуется принципом последовательного изменения состояния операнд итерационным образом, ориентированная на классическую модель *Джона фон Неймана:*
1.1.1. Метод последовательного изменения состояний, поддерживаемый концепцией алгоритма;
1.1.2. Метод потокового управления на исполнение, в итерационном контроле.
1.2. Методология объектно-ориентированного программирования (ООП), которая использует объектные декомпозиции. Отметим, что статическая структура ИС и сред описывается в терминах объектов и связей между ними, а поведение системы описывается в терминах обмена сообщениями между объектами, как в UI/UX и системном программировании.
Пример: инкапсуляция, то есть абстрактный тип данных, его наследование и полиморфизм, с применение таких методов как:
1.2.1. Метод объектно-ориентированной декомпозиции, который заключается в выделении объектов и связей между ними, поддерживающийся концепциями инкапсуляции, наследования и полиморфизма;
1.2.2. Метод абстрактных типов данных, который лежит в основе инкапсуляции, поддерживающийся концепцией абстрагирования;

Классификация Secure SDLC, Часть 2

1.2.3. Метод пересылки сообщений, который заключается в описании поведения системы в терминах обмена сообщениями между объектами, поддерживаемый концепцией сообщения.
1.3. Методология функционального программирования – как способ составления программно-аппаратной части, в которой единственным действием является вызов функции, как вариативы, то есть способа расчленения программно-аппаратной части на отдельные сектора, где имеет место быть формат введения имени для функции и задания для него вычисляющего значения, применяемый с такими методами концепций как:
1.3.1. Метод аппликативности, где: программно-аппаратная часть — есть выражение, которое подставлено из функции к аргументу, состоящему из определения функции, представляющей собой вызов от другой функций и вложенной друг в друга, поддерживается концепцией функции;
1.3.2. Метод рекурсивного поведения, который заключается в самоповторяющемся поведении, то есть возвращающемуся к самому себе, поддерживается концепцией рекурсии;
1.3.3. Метод настраиваемости, который заключается в порождении новых программных объектов по специальному образцу, где значения соответствующих выражений применяется как порождающая функция к параметрам образца.

Классификация Secure SDLC, Часть 3

1.4. Методология логического программирования, где программно-аппаратная часть содержит описание проблемы в терминах фактов и логических формул, а решение проблемы ИС и среды выполняется с помощью механизмов логического вывода, где применяются такие методы и концепции как:
1.4.1. Метод единообразия, где используется применение механизма логического доказательства ко всей программе;
1.4.2. Метод унификации, то есть механизм сопоставления с образцом для создания и декомпозиции структур БД.
1.5. Методология программирования с ограничениями, при котором в ПО определяется тип данных для его решения, где предметная область шифруется на соответствующие ограничения значений искомого решения, которая находится в ИС и средах. В данной методологии предлагается двухуровневая архитектура, которая интегрируется как компонент ограничения программно-аппаратного компонента. В данном формате подразумевается описательная модель вычислений, где программа содержит описание понятий и задач в формате поддерживания концепции модели.

Классификация Secure SDLC, Часть 4

Классификацию по топологической специфике самой методологий — то есть форм-факторы топологии на базе методологии со способностью выбора самих методов для получения уточненного ядра, с такими методами и концепциями как:
2.1. Последовательная декомпозиция алгоритма решения задачи сверху вниз – в итерационной детализации, которая начинается с общей задачи и обеспечивает ее структурированность, которая поддерживается концепцией алгоритма;
2.2. Метод модульной организации частей программы, где происходит разбиение программы на специальные компоненты, которые поддерживаются концепцией модуля;
2.3. Использование структурного кодирования, где используется кодирование трех основных управляющих конструкций, которые поддерживаются концепцией управления.

Классификацию по реализационной специфике методологии, где применяется использование каждого из ядер методологии с конкретной спецификой. В данной классификации определяется некоторая организация аппаратной поддержки, такая как: централизованная или параллельная, использующаяся для централизованных архитектур;

Классификацию по смешанной методологии, которая включает объединение методов нескольких методологий, таких как методологии функционального и логического программирования;

SUNLIGHT

Классификация Secure SDLC, Часть 5

Классификацию по идейной задумке *Петрова В.Н.*: "Технологии разработки программного обеспечения", являющейся методологией – RAD (Rapid Application Development). Данная методология носит наименование — методологии быстрой разработки приложений. Целевая предпосылка в области инструментальных средств быстрой разработки приложений основана на таких элементах как:

5.1. Объёмное положение разработчиков, которые разрабатывают АИС со всей спецификой и вариацией относительно ТТ, ТЗ. Например: минимальная группа разработчиков, представляет из себя от 2 до 10 человек, для большей части вариатива, может достигать до 100 разработчиков, такая специфика прослеживается в основном в игровой индустрии;

5.2. Тщательная проработка производственного графа работ кадрового состава организации, его оптимизации и управления над ним, так же влияющий на календарный план разработки и прототипирования. Данный граф по временным отрезкам представляется от 2 до 6 месяцев, в среднем, но все зависит от целевых нужд организации для АИС, согласно ТТ и ТЗ.

Принципы выбора методологий в безопасной разработке

- Четкое понимание необходимого и конечного функционала, дизайна, прототипа интерфейса UI/UX и всей грядущей разработки АИС, то есть присутствует детализация специфики АИС. Стоит отметить, что она является четко регламентированной, с конечным конкретным ТЗ и ТТ, где описано: что и как должно работать;
- Формализованные целевые нужды конечного потребителя, тогда когда разработчики и руководящий состав, включая Заказчика представляют четкую "картину" в документированном виде, то есть: создается подробное ТЗ, где данная дефиниция регламентируется полноценным процессным состоянием каждого элемента в АИС, которая также содержит полноценное описание в цикле по специализированным алгоритмам, включая детализацию по UI/UX;
- Конечные требования к АИС являются стабильными, вследствие чего потребитель не дополняет условий по изменению функциональности АИС во время ее разработки, модернизации, исключая возможный формат блочного масштабирования в рамках текущих отношений;
- Представление итерационного процесса в формации аналитики, где проектирование и разработка ТЗ строго линейно.

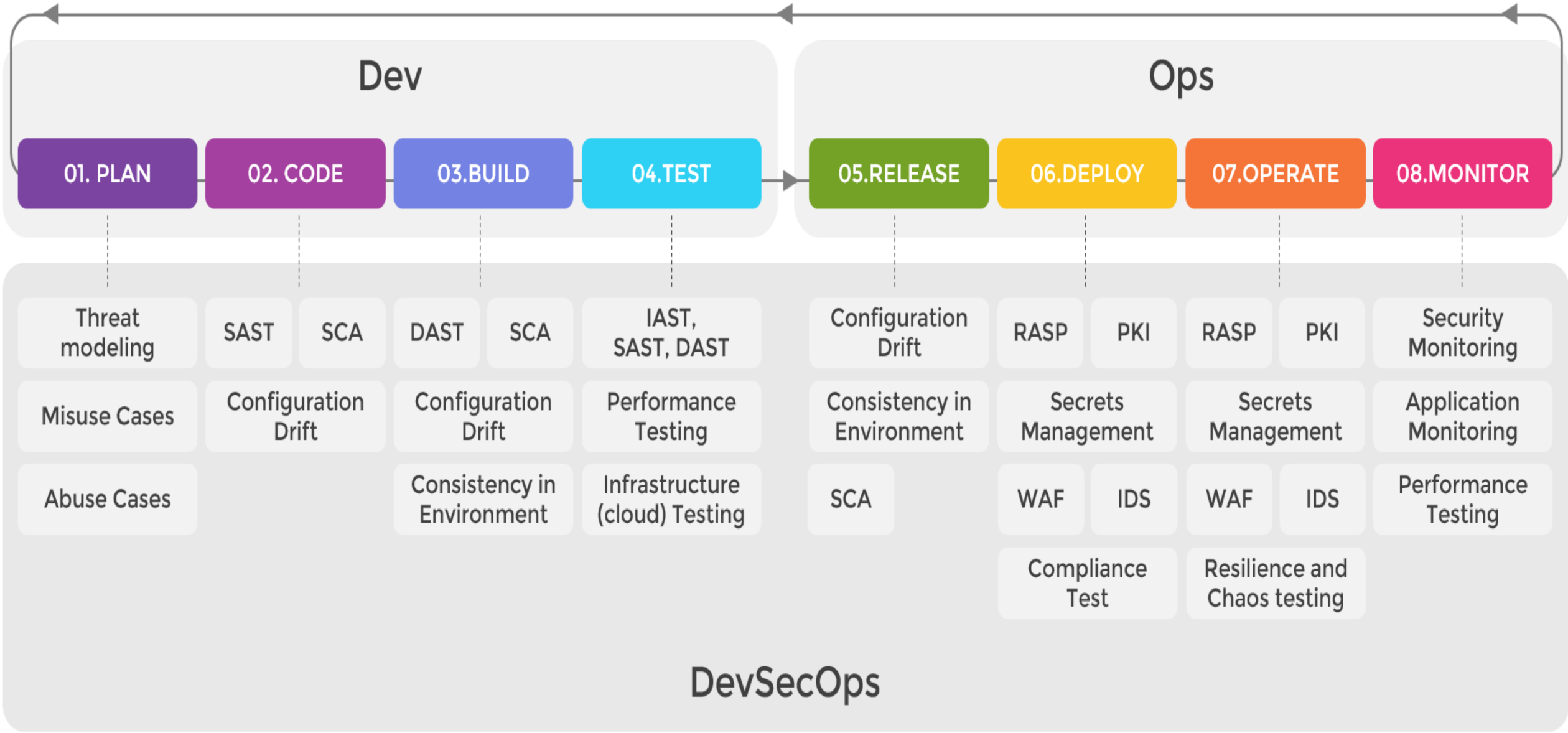
SUNLIGHT

Адаптивная методология Secure SDLC

- Предоставляются понятийные, не изменчивые требования к АИС;
- Потребителю предоставляется разрабатываемая АИС только в общих очерках, где предполагается внесение изменений функциональности или дизайна разрабатываемой АИС посредственно в сам момент произведения процесса разработки;
- Представление необходимости быстрого получения первых вариаций версионности продукта и дальнейшего масштабирования исходя от нужд организации и решений *Digital*, в момент работающего АИС;
- Представление решаемой задачи посредством программно-аппаратного комплекса АИС неэффективно поддается документированию по объективным причинам;
- Представление реализации всех требований к проекту, вновь внесенных в том числе, которые варьируется как добавочный ликвидный запас временного отрезка для разработки и прототипирования.

SUNLIGHT

Secure Software Development Life Cycle (SSDL)



SUNLIGHT

Прогнозирование адаптивной методологии

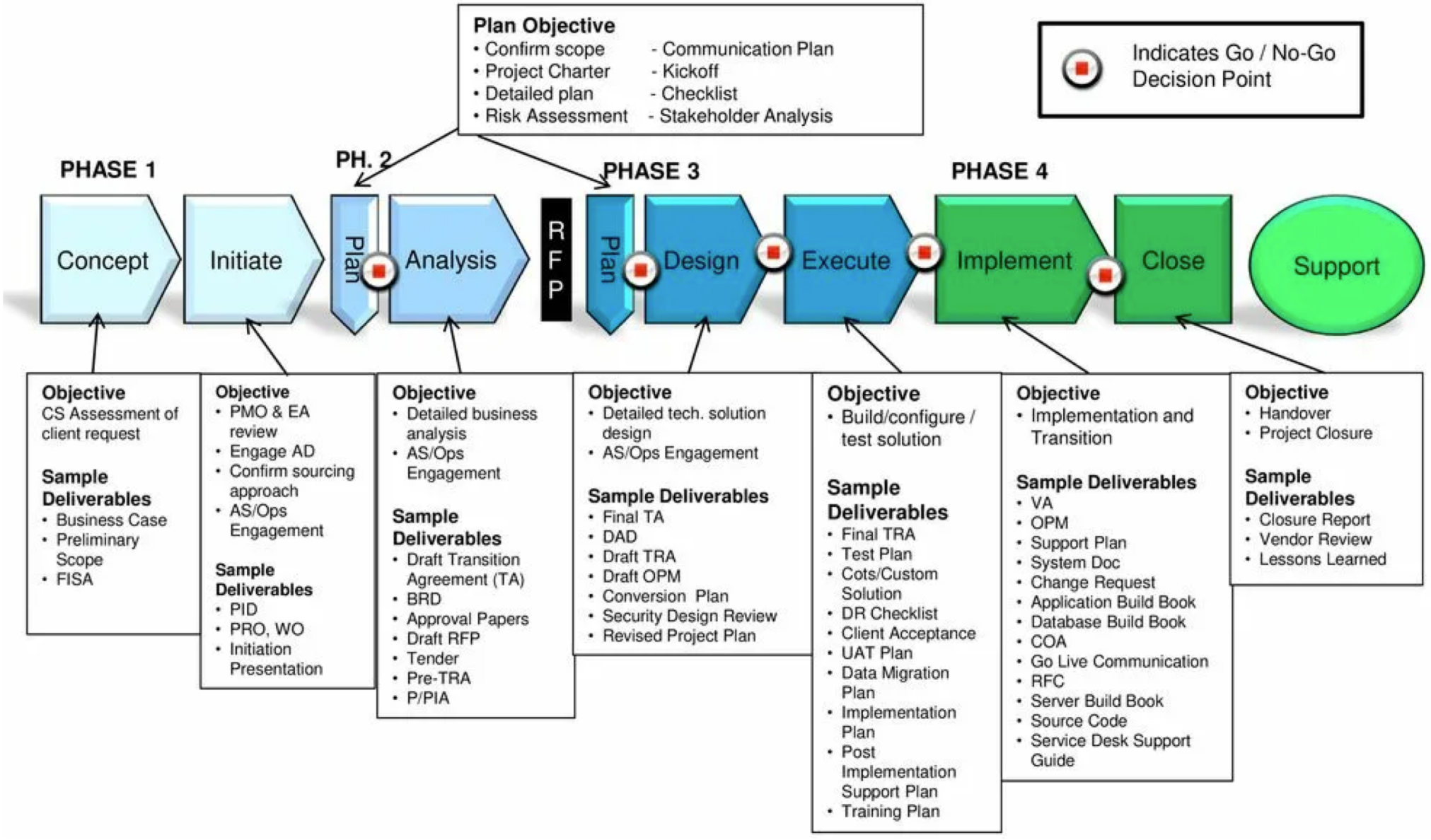
- Приведенная методология позволяет спрогнозировать разностороннюю вариативность при разработки и прототипировании продукта, исходя из общего формата различных пользовательских кейсов, в том числе со стороны злоумышленник по типу: "а как бы это сделал я?".
- Методология позволяет предопределить возможные вариативные изменения в процессе разработки и контролировать итерационность получаемого продукта, что позволяет "бизнесу" контролировать риски, в том числе со стороны финансирования организацией.
- Отмечу, что приведенные принципы схожи с возможностью изменения продукта исходя из метрик при анализе для "бизнеса", что позволяет в процессе разработки контролировать все жизненные циклы и процессы версионности продукта, в том числе монетизирования.
- Этот формат подходит для долгосрочных продуктов, которым необходимо иметь возможность постоянной монетизации и частично быть конкурентноспособными исходя из максимизации перекрытия спроса — предложением.

SUNLIGHT

Выводы, Часть 1

- Стоит отметить, что разработка и проектирование ПО при безопасной разработке основывается на базе функциональности ИС и совокупных сред, спроектированных относительно конечного ТТ и ТЗ.
- Также не мало важно понимание руководящего состава организации в обоснованности и целях безопасной разработке, где весомым показателем в данной ситуации являются оценочные метрики бизнеса в отношении рисков, которые фокусируются на целевых функциях ПО для монетизации.
- Данные функции разрабатываются и интерпретируются на вывод в необходимом количественном и качественном формате сведений, посредством представленных систем и подсистем в СУИБ, а не в процессе их верификационного построения.
- Приведены конечные форматы смежных политик ИБ на базе методологий безопасной разработки ПО, включая методов и средств управления, включая оценочные метрики бизнеса под определенными видами нагрузок, стрессовых режимов по БП, а также в виде количественных и качественных характеристик обрабатываемой информации.

Выводы, Часть 2



SUNLIGHT

Sunlight SDLC и продуктовые решения

SUNLIGHT — БОЛЬШЕ ЧЕМ ЮВЕЛИРНАЯ КОМПАНИЯ

5 500	156	120
СОТРУДНИКОВ	ГОРОДОВ	ПРОИЗВОДИТЕЛЕЙ
70 000	400	25млн
ЮВЕЛИРНЫХ УКРАШЕНИЙ	МАГАЗИНОВ	ПОКУПАТЕЛЕЙ

E-commerce
Разработка высоконагруженных систем с миллионами пользователей и сотнями тысяч заказов в месяц

Supply Chain
Управление IT-системами, которые обеспечивают непрерывные поставки и быструю доставку

Мобильная разработка
Мобильные приложения для IOS и Android: первые места в сторах и миллионы уникальных пользователей

BI&Big Data
Десятки миллионов событий каждый день, современные продукты для аналитики

Marketplace и портал поставщиков
Крупнейший ювелирный маркетплейс и платформа для работы с поставщиками

Retail-платформа
Платформа для магазинов на базе 1С, в которой ежедневно совершаются миллионы операций

SUNLIGHT — это:

- сайт и мобильное приложение, которые посещают более 30 миллионов раз в месяц;
- приложение для продавцов, которым пользуются 4000 продавцов ежедневно - SUNRETAIL — собственная разработка отдела IT & DIGITAL.
- WMS, в котором собираются десятки тысяч заказов каждый день;
- розничная платформа, обрабатывающая миллионы событий каждый день, а также развитая отказоустойчивая IT инфраструктура с сотнями физических и виртуальных серверов.

За 20 лет мы выросли из ювелирного бренда в технологическую компанию!

Сайт и мобильное приложение SUNLIGHT — мощная IT-структура, с сотнями физических и виртуальных серверов.
Это уникальное приложение, которое уже несколько лет успешно оптимизирует работу более 4 тыс. человек в розничном секторе компании.

SUNLIGHT

Спасибо за внимание ;)
Вопросы?