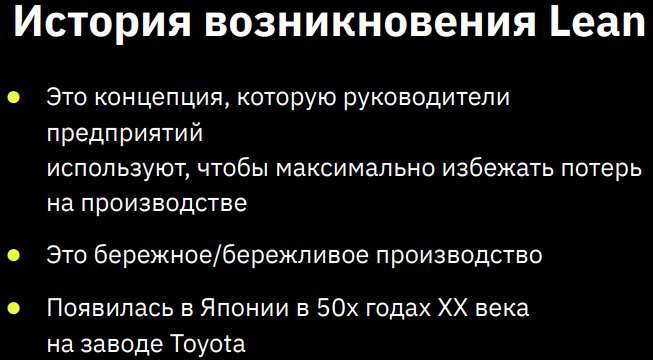
Лекция № 4. Lean

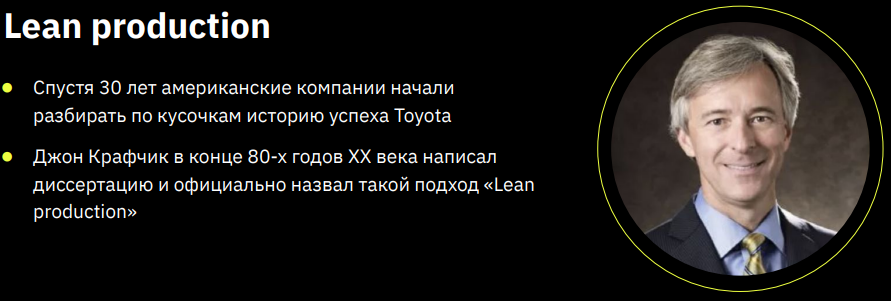
Сегодня мы поговорим про Lean, про такую интересную концепцию, подход к разработке it-продуктов. Про что мы будем сегодня говорить, во-первых, обсудим. Что такое Лин? Как он применяется? Какие включает в себя принципы, ценности? Естественно, посмотрим всё это на примерах, как это внедрить в IT? Какие есть пошаговые инструменты, для внедрения данной концепции в IT? Что такое Lean? Как появился?



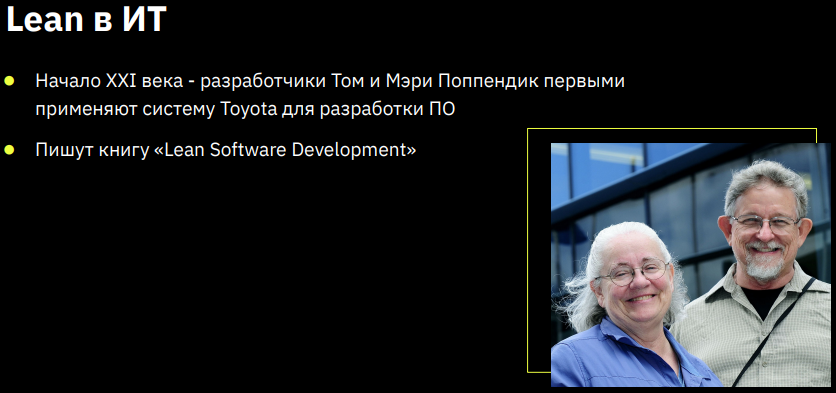
По сути дела, лин — это концепция, которую применяли на производстве в компании Тойота. Что это такое? Это бережное, бережливое производство автомобилей. Конечно, уже в Тойоте разрабатывали автомобили и применили, поэтому там подходили с точки зрения бережливого производства автомобилей на конвейер. Появилось это, в Тойоте в Японии, в пятидесятых годах XX века, компания в то время столкнулась с серьёзными проблемами, реализации своих автомобилей. Почему? Потому что очень сильно возросла стоимость, производства каждого автомобиля, компания не могла конкурировать с большими крупными автомобильными гигантами, в то время американский рынок осваивал Ford. Япония пыталась выйти на зарубежный рынок, потому что в стране, была достаточно серьёзная, сложная, ситуация экономическая, но тем не менее из-за высокой стоимости автомобилей, у ней это не получалось сделать.



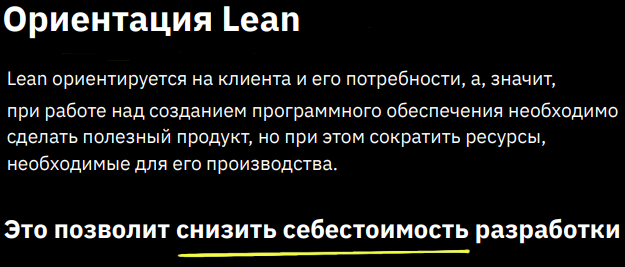
Что они решили предпринять? Тайити Оно и Сигео Синго, это промышленные инженеры и предприниматели, решили применить подход бережного производства к автомобилю. Они научились, в то время, быстро перенастраивать конвейер, производство одной марки на другую, в то время, у многих компаний уходила на это несколько дней, а то и недель. Это помогло компании встать, из маленькой компании, превратиться, полностью в большого автомобильного гиганта, про который мы сейчас естественно все знаем.



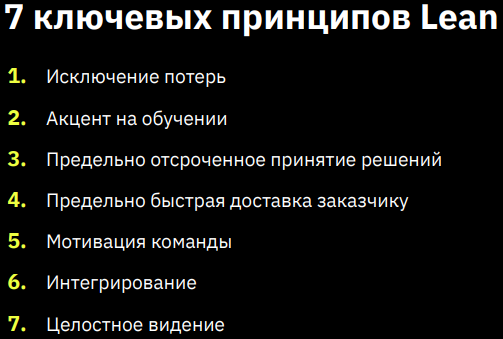
Спустя 30 лет американские компании начали разбирать по кусочкам историю успеха Toyota и что они выявили, был такой человек, зовут его Джон Крафчик, в конце восьмидесятых годов XX века, он написал диссертацию и официально назвал, такой подход, как Лин продакшн, то есть бережное производство. А как вошёл Лин в IT? Как применить концепцию в IT сфере?



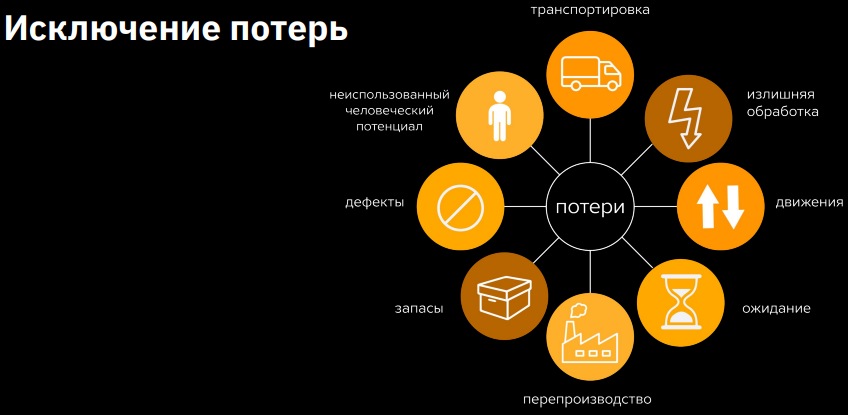
Давайте поговорим, в начале XXI века были такие разработчики их звали Том и Мэри Поппендик, они впервые применили данную концепцию, при разработке ПО. Затем они написали книгу, лин софтвер девелопмент, в которой уже полностью изложили, все принципы, подходы и как это можно применить, при разработке продуктов.



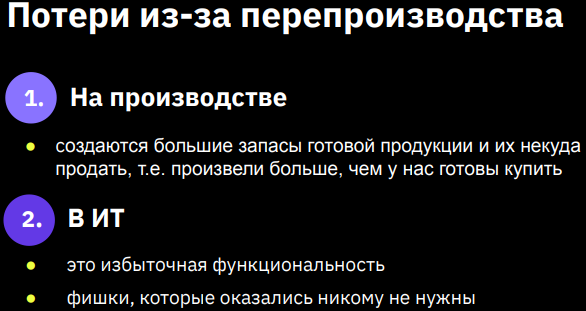
Важно отметить, что эти два разработчика, поняли, что ориентация лин направлена на то, чтобы создать, конечный качественный продукт и удовлетворить, все потребности, нашего заказчика, но сделать это, с помощью сокращения затрат, сделать разработку более эффективной. Конечно же учитывая, что это позволяет снизить, себестоимость разработки нашего it-продукта, это повышает нашу конкурентоспособность на рынке, компаний разработчиков, какие есть принципы лин и как их применить в практике? Давайте посмотрим, есть 7 ключевых принципов лин, первое — это исключение потерь, основа основ, на основе которой строится вся данная концепция, второе — это акцент на обучение, третье — это предельно отсроченное принятие решений, предельно быстрая поставка заказчику, мотивация команды, интегрирование, постоянное и конечно же, целостное видение, всего процесса.



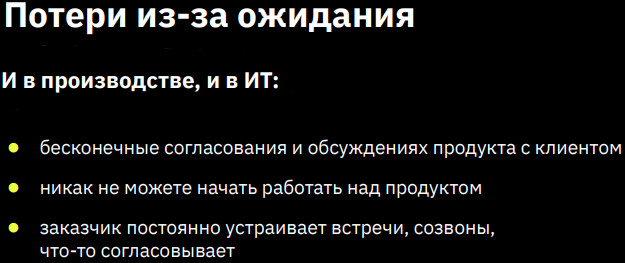
Давайте по порядку, смотреть каждый из данных принципов, исключения потери,



здесь вы видите все потери, которые возможны при разработке продукта, будь то it-продукт, либо производство автомобилей, или другого какого-то оборудования. Это потери, при производстве, при ожидании, при движении, при излишней обработке, от транспортировки, каких-то дефектов, и много ещё, чего другого. Давайте смотреть последовательно, каждую из данных потерь.



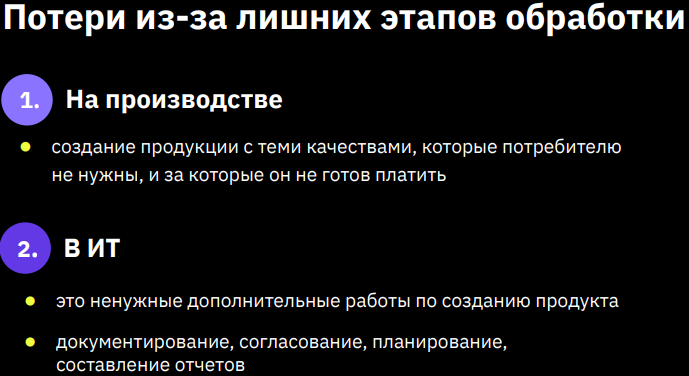
Потери из-за перепроизводства, касательно производства — это создание больших, запасов готовой продукции, то есть мы произвели больше, чем у нас готовы купить, если мы говорим про IT, то это избыточная функциональность. Допустим мы разрабатывали какой-то продукт, какую-то программу, и сильно увлеклись, разработали очень много. Конечно же классных, хороших интересных фишек, но, к сожалению, заказчику они не нужны, заказчик не готов за них нам заплатить.



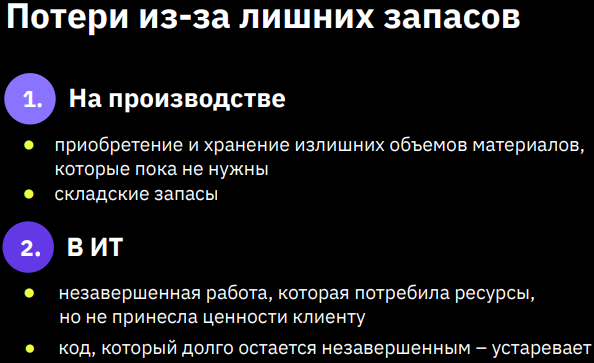
Потери из-за ожидания, и в производстве, и в IT, они очень схожи, это потеря времени, когда мы никак не можем начать работу. Мы постоянно, что-то согласовываем, пере согласовываем, получаем новые исходные данные, меняем, какие-то процессы, в целом, это всё тормозит работу. Либо мы начинаем, какие-то излишние обсуждения нашего продукта с заказчиком, заинтересованными сторонами, важно понять определили ли требование перешли к разработке и соответственно делаем работу, не теряем зря время.



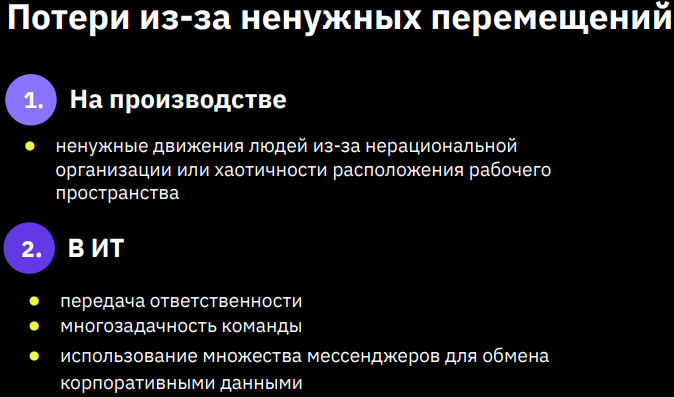
Потери при ненужной транспортировке, на производстве — это происходит, когда у нас не отлаженные процессы логистики, когда в целом очень много различных звеньев, через которые происходит транспортировка, каких-то запасных частей, изделий готовой продукции, и так далее. Если мы говорим про IT, это передача данных между, большим количеством людей и систем, опять же нарушения в процессе. Основная цель — это сделать, передачу данных внутри компании разработчика, быстрой, эффективной и без лишних потерь, на ненужные звенья.



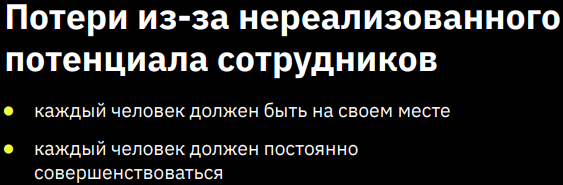
Следующие потери, это потери из-за лишних этапов обработки, на производстве — это создание продукции с теми качествами, которые не нужны нашему конечному заказчику, и соответственно, он за них не готов платить, если мы говорим про это IT, то это не нужно. Дополнительные работы, с которыми мы сталкиваемся при разработке продукта, допустим ненужное согласование, ненужное планирование, составление каких-то документов, которые не несут в себе практической ценности, отчётов и так далее, избавляемся от всего ненужного.



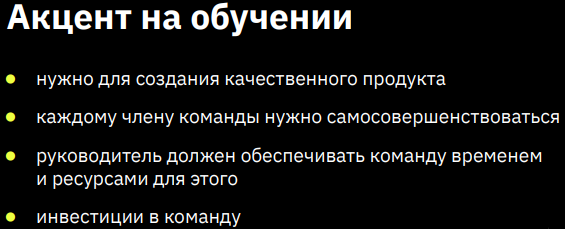
Следующие потери, это из-за лишних запасов, на производстве — это конечно же, хранение больших объемов материалов, тех самых запасных частей, которые ещё не поступили в работу, не поступили в производство, те самые наши складские запасы, которые тратят наши деньги, и делают неэффективно, в целом производство. Как раз, это та проблема, с которой столкнулась Тойота. В IT, это незавершенная работа, какой-то код, который долго остается незавершенным, он устаревает, и важно отметить, что, когда высота разрабатываете, пишите код, он должен, либо перейти затем уже в ПО, либо его, не нужно писать вообще.



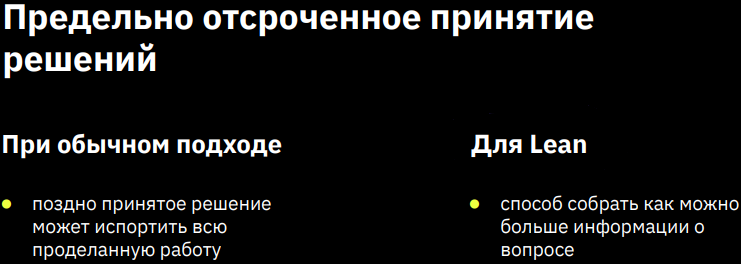
Потери из-за ненужных перемещений, на производстве, ненужные перемещения у нас возникают из-за нерациональной организации или хаотичности, в целом, расположения рабочего пространства. Если мы говорим про IT, то здесь происходит ненужная передача ответственности. Допустим вы определили, уже закрепили, определённые задачи за людьми, а затем по мере того, как они начали над ними работать, принимаете решение перекинуть эту ответственность на других людей. Зачем? Не нужно этого делать, если люди с этим хорошо справляются всё хорошо, пусть продолжают. Также важный момент, когда происходит многозадачность команды, когда вы ставите, очень много разных целей, задач своей команде. Естественно, людям нужно переключаться с одного момента на другой, либо с одного продукта на другой продукт, и в этом происходит потеря, как раз-таки, из-за ненужного перемещения, пока мозг человека переквалифицирует, перестроиться, так сказать на другой процесс, мы потеряем время. Поэтому важно, опять же, не нагружать сильно свою команду, всё по возможностям. Ещё один интересный момент, когда компания начинает использовать несколько различных мессенджеров, для коммуникации, допустим это WhatsApp, Telegram, Viber, ещё есть какие-то, внутренние ресурсы в компании и так далее. Зачем, нужно делать поток информации предельно понятно, и лучше всего соблюдать, единство, то есть передача информации должна происходить, в каком-то одном мессенджере, потому что, по прошествии определённого количества времени, мы даже, можем не вспомнить всю цепочку, и не понять, откуда возникла какая-то проблема, и на каком этапе, она у нас появилась. Потери из-за дефектов, здесь всё очень просто, это дефекты, это баги, при написании кода, при отрисовки дизайна, и так далее. Важно обучать свою команду, важно уходить, от этих самых потерь, которые мы тратим на устранение наших дефектов, на устранение багов, это время. Время для нас, это деньги.



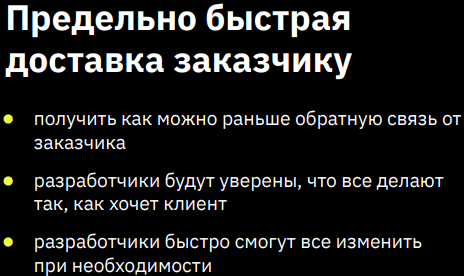
Потери из-за нереализованного потенциала сотрудников. Очень важный момент, чтобы все люди находились у нас на своих местах, потому что нет ничего хуже, чем нереализованный потенциал человека, если допустим вы нагружаете своего специалиста однотипными задачами, долгое протяжённое количество времени, то ему элементарно становится неинтересно выполнять эту задачу, несмотря даже на его опыт, на его квалификацию, он может просто загрустить, поэтому важно проводить определённую ротацию, и повышать квалификацию людей.



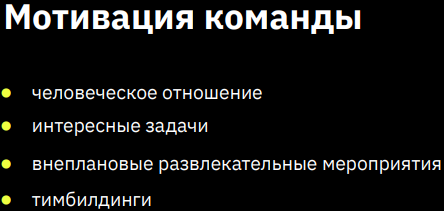
Следующий принцип, акцент на обучении, как раз-таки, мы говорили про потерю из-за нереализованного потенциала человека, важно делать акцент на обучение, важно совершенствовать своих людей в команде и именно как руководитель, давать им возможность. Для этого, надо предоставлять ресурсы, предоставлять время, предоставлять поддержку, для того чтобы они обучались, совершенствовались и приносили какие-то улучшения в разработку продукта. Это определённые инвестиции, долгосрочные инвестиции в вашу команду.



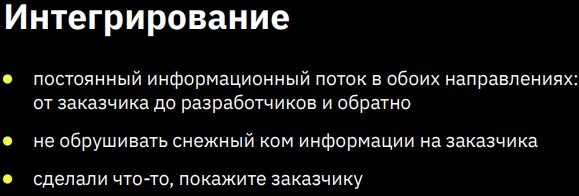
Следующий принцип, это предельно отсроченное принятие решений, если мы берём обычный подход, то там, поздно принятое решение, может всё сломать, она может очень дорого для нас стоить. Если же мы говорим про лин, то тут немножко всё по-другому, здесь отсроченное принятие решения — это способ собрать гораздо больше информации о вопросе, и не допустить ошибку.



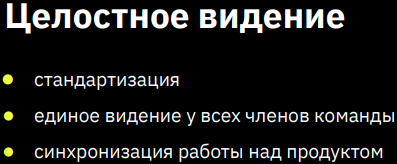
Предельно быстрая доставка заказчику, важно отметить, что вам как разработчикам, нужно получить, как можно быстрее, обратную связь от заказчика, сделали показали, если всё хорошо, перешли на следующую часть, если есть какие-то дефекты или баги, исправили и продолжаете дальше работу. Не нужно ни в коем случае сбрасывать, одновременно все элементы сразу же на заказчика, делайте постепенно.



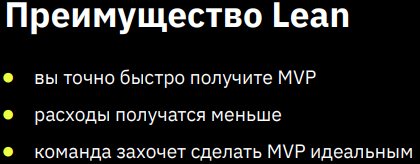
Мотивация команды, важный аспект — это те самые тимбилдинги, это хорошие отношения в коллективе, это интересные задачи, которые ставит руководство, и в целом человеческие взаимоотношения, отношения между членами команды разработки.



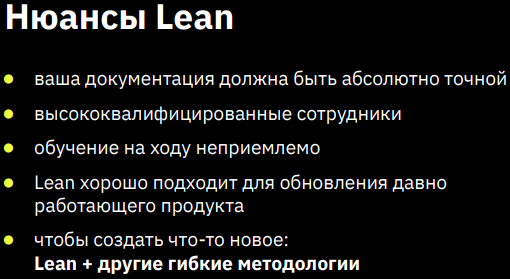
Интегрирование. Интегрирование, предполагает наличие постоянного, непрерывного информационного потока, от заказчика к разработчикам, и в обратную сторону, то есть, очень похож на предыдущие принципы, когда мы говорили, что необходимо делать всё поэтапно, то есть, не обрушивается снежный ком, на своего заказчика, поставляйте элементы своего продукта, частями, поэтапно, получайте обратную связь и улучшайте свою работу.



Целостное видение. Здесь основной акцент, необходимо сделать, на работу руководителя разработки, либо руководителя команды, или, неважно как он называется в вашей компании. Важно сформировать, общее, целостное видение конечного продукта, то есть выработать определённые стандарты, стандартизировать работу, чтобы всем было, предельно понятно, всем членам команды, предельно понятно, над чем они работают, и у них было единое видение конечного продукта. Учитывая все вышеперечисленные принципы, Лин, обладает преимуществами и нюансами, именно нюансами, не недостатками.



Какие преимущества вы получите, от применения Лин концепции, конечно же вы достаточно быстро получите минимальную версию продукта (МВП), вашего, но с гораздо меньшими потерями, с меньшими расходами, и в любом случае ваша команда захочет сделать эту МВП, эту вашу минимальную версию продукта идеальной.

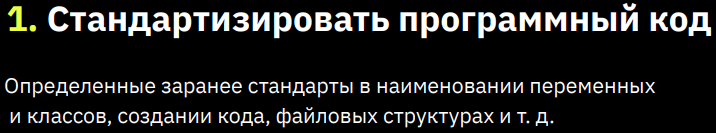


Но есть ещё и следующие нюансы, которые важно учитывать, если вы хотите применить данную концепцию. Первый основной нюанс, заключается в том, что ваша документация идеальна, то есть она должна быть технически, достаточно проработана, и точно. Второй момент, в вашей команде, должны быть только высококвалифицированные люди, потому что обучение на ходу, ни в коем случае неприменимо, иначе вы потеряете, потеряете на дефектах, произойдут те самые потери, про которые мы говорили. Лин достаточно хорошо подходит для обновления, какого-то уже работающего продукта, а чтобы создать что-то новое, вам нужно лин строить, в какую-то другую гибкую методологию разработки. Это очень важный момент, сама по себе, лин концепция не существует, для того чтобы внедрить лин концепцию, существуют определённые инструменты бережливой разработки.



Давайте, поговорим про них подробнее, во-первых, если вы хотите внедрить лин концепцию, по разработке it-продукта. Вам необходимо, проанализировать и найти приоритетные направления, которые вы хотите поработать и усовершенствовать. Сама лин концепция работает таким образом, что она позволяет, при меньших затратах, получить быстро, программный продукт. То есть, если мы говорим, про какие-то другие гибкие методологии, например скрам, там есть чётко выделенные, определённые, порядок действий, определённый набор этих действий, определённые принципы и практики, которые нельзя менять. Лин же пытается, максимально быстро, поставить законченный продукт, при этом с наименьшими потерями, то есть исключаем все потери, и работаем, концентрируемся на создание продукта.

Какие есть практические шаги к внедрению Лин в вашей компании, при разработке?



Во-первых, это стандартизировать программный код, определите заранее, основные стандарты, наименованиях, переменных, классов, и так далее, то есть полностью стандартизируйте работу, как вы будете, что называть.



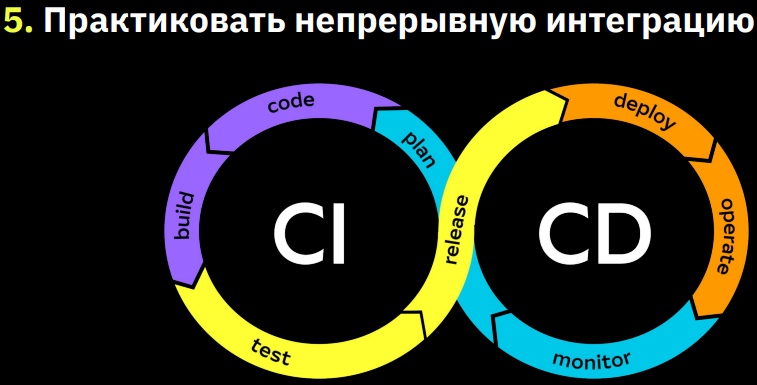
Второй шаг, это использование парного программирования, очень интересный метод, когда за одним компьютером, код пишут два человека. Здесь такой подход, что одна голова хорошо, а две лучше, соответственно, это как определенный штурман, который подсказывает дорогу водителю. Исключаем те самые потери при дефектах, то есть уже у нас происходит такой своеобразный парный контроль, над написанием кода.



Третий шаг, это встраивать защиту от ошибок, такая защита, от дурочка, создавайте интерфейс, создавайте код таким образом, чтобы люди не могли, случайным образом, внести какое-то некорректное изменение в него, не могли допустить ошибку. Поэтому делайте всё предельно понятно, ясно, чтобы у человека не возникало каких-то вопросов, не нужных из-за, которых мы соответственно, потеряем, время и ресурсы.



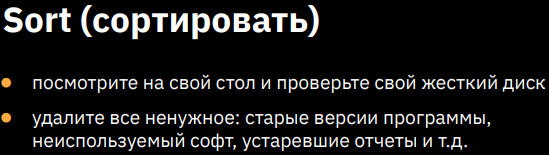
Автоматизировать рутинные задачи, очень важный момент, напишите скрипты по сборке, автоматизируйте абсолютно любую рутинную работу, которая у вас встречается на проекте, чтобы исключить потери времени.



Практикуйте непрерывную интеграцию, это очень интересная it-практика, CI и CD (Continuous integration и Continuous Delivery), это происходит тогда, когда вы написанный код внедряете в какое-то тестовое пространство ПО, для которого соответственно пишите что-то. Очень важный момент, заключается в том, что внедрять лучше всего в тестовое пространство, не сразу в рабочее, чтобы не полетели какие-то базы, не слетела работа в целом, вашей компании.

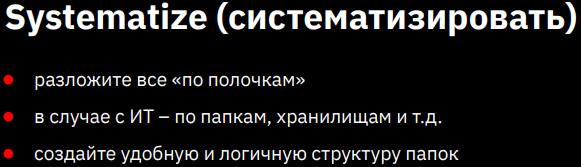


Также, для внедрения Лин в вашу работу, есть интересная японская практика, называется она 5S. Давайте посмотрим из чего состоит данная японская практика, 5S — это первые пять букв с от слов,

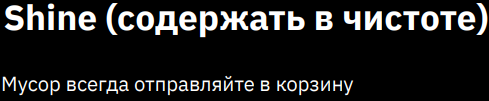


во-первых, сортировать, систематизировать, содержать в чистоте, стандартизировать, и конечно же существовать, и поддерживать.

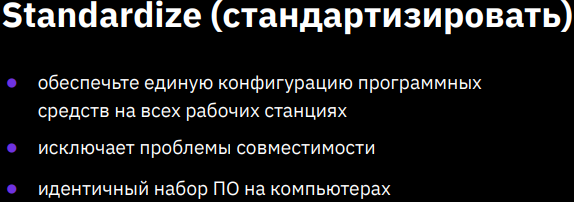
Давайте по порядку. Что такое сортировать? Посмотрите, на свой стол, на свой жёсткий диск, посмотрите, какие у вас есть ненужные, устаревшие элементы, и полностью избавьтесь от них, то есть блокируйте.



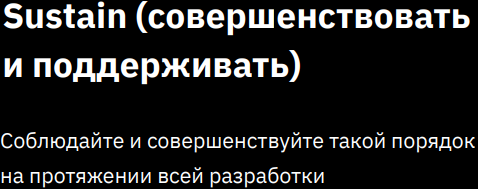
В следующий момент, это систематизировать, затем выстроить, определенную структуру папок, внутри хранилища, внутри вашего жёсткого диска. Неважно, где главная, систематизируйте полностью иерархию всех файлов на вашем, допустим компьютере, на вашем хранилище, либо в вашей работе. Это очень важный момент, то есть необходимо создать удобную и понятную иерархию.



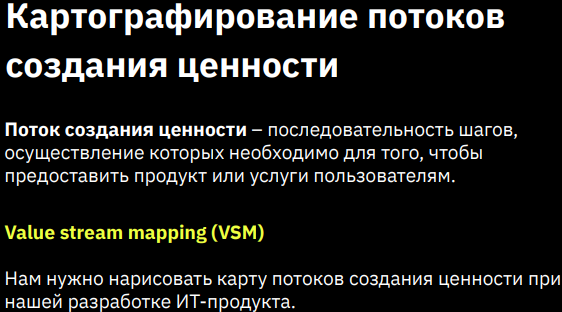
Следующий момент, это содержать в чистоте, все ненужные, старые элементы, отправляйте в корзину. Конечно, если необходимо что-то заархивировать, создать резервную копию, делайте. Но избавляйтесь, от всего ненужного, очищайте своё рабочее пространство.



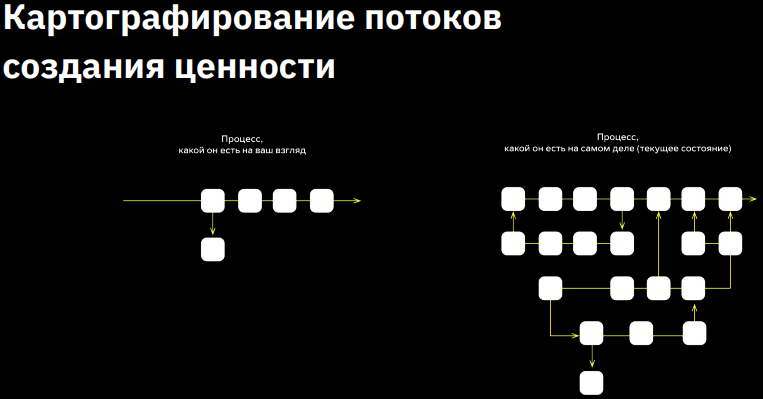
Затем стандартизируйте, то есть обеспечьте, единую конфигурацию программных средств, на всех рабочих станциях. Важно отметить, что данный подход позволяет исключить проблемы совместимости, какого-то кода, какого-то ПО и так далее, то есть должен быть идентичный набор ПО на абсолютно всех компьютерах, которые задействованы при разработке продукта, и после того, как вы все, эти четыре шага пройдёте,



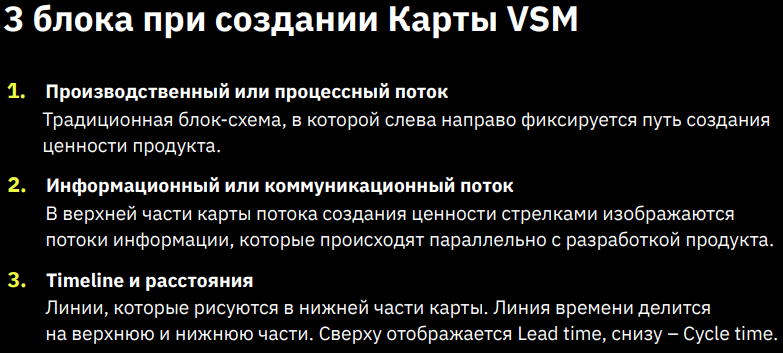
вам необходимо совершенствовать и поддерживать, такой выработанной порядок. Очень интересный и важный момент — это картографирование потоков, создания ценности. Что это такое, когда мы хотим, применить Лин при разработке продукта? В целом, важно понимать, что поток создания ценности, это последовательность шагов, которые мы выполняем для того, чтобы предоставить, то есть разработать наш продукт.



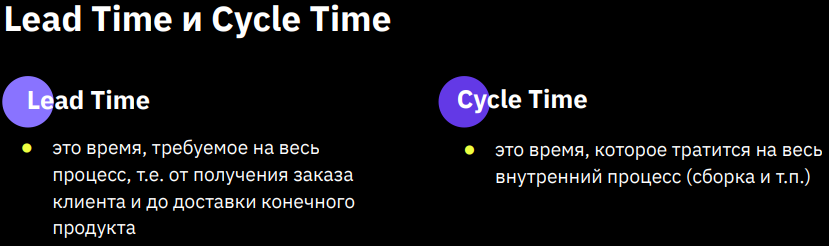
Соответственно, нам необходимо отрисовать, эту самую карту процессов, которые у нас происходят при разработке продуктов, она ещё называется Value Stream Mapping (Карта Потоков Создания Ценности). Это очень важный момент, и необходимо на нём, заострить внимание, рисуем карту потока создания ценности, при разработке нашего it-продукта.



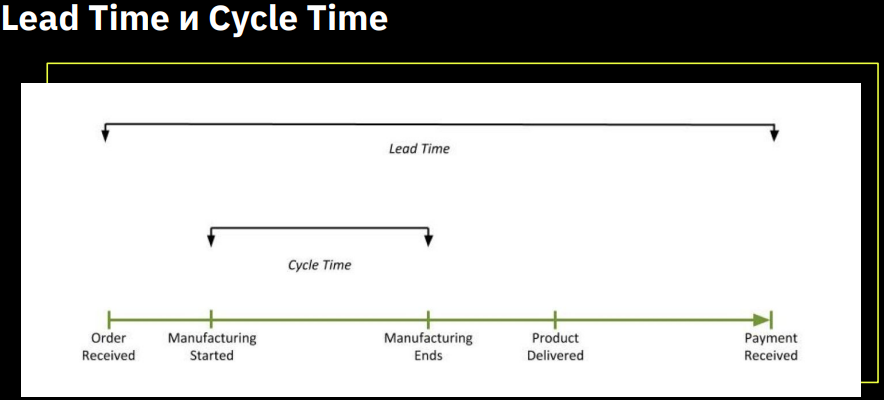
Давайте посмотрим, здесь вы видите, на данном слайде, слева, это процессы: «Как вы их представляете себе? То есть, как вам кажется, что происходит при разработке? Что происходит в вашей компании допустим?». А справа, это процесс, какой есть на самом деле, то есть он очень завихрённый, очень запутанный, какие-то стрелки, какие-то квадратики, и так далее. То есть, очень много различных звеньев, до того момента, пока вы получите готовый продукт. Для того, чтобы нарисовать, эту самую VSM, эту самую, карту потока создания ценности, нужно сделать три важных шага.



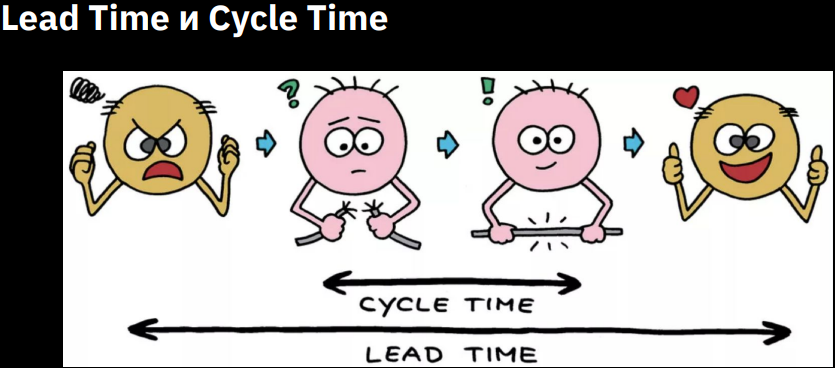
Первый шаг нарисовать производственный или процессный поток, то есть, это традиционная блок-схема, в которой у нас показано стрелочками, откуда, что, слева направо, каким образом, фиксируется весь путь создания нашей ценности, затем мы рисуем, информационный или коммуникационный поток, то есть показываем, откуда, из каких каналов мы получаем информацию, для выполнения определённых работ, на нашем проекте и третий важный шаг, это тайм лайн (Timeline), расстояние те самые сроки, все те самые периоды, в рамках которых, мы что-то делаем, при отрисовке тайм лайна, важно понимать, что сверху. Здесь вам нужно отобразить лит тайм (Lead time), а снизу сайкл тайм (Cycle time).



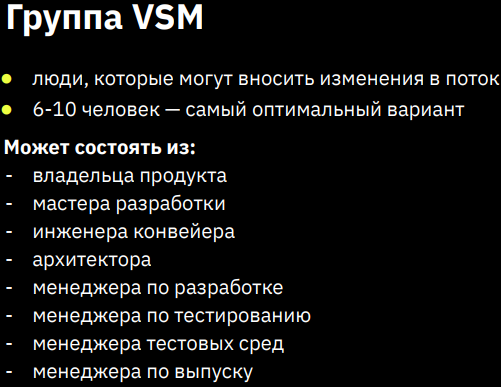
Что это такое? Давайте посмотрим, лит тайм — это время, которое у вас затрачивается на весь процесс, то есть от получения заказа клиента до получения оплаты за него, а сайкл тайм — это время, которое затрачивается именно на разработку, какой-то части, либо, полностью всего нашего it-продукта,



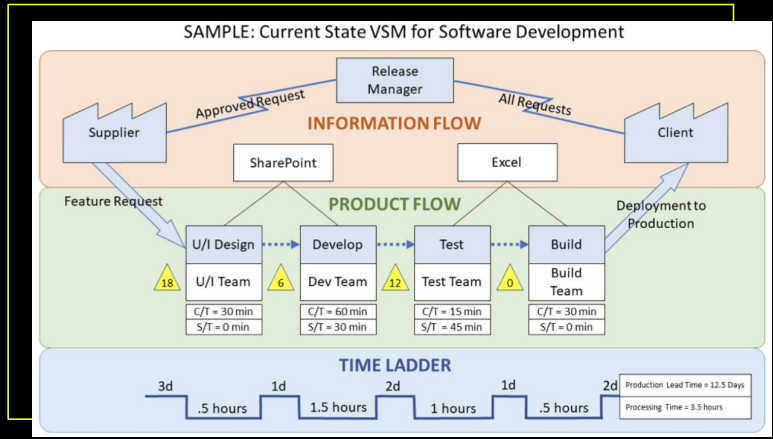
если смотреть графически, то здесь вы видите, что лит тайм он гораздо длиннее, чем цикл сайкл тайм, то есть сайкл тайм – это, тот период времени, которое мы затрачиваем на производство, какой-то части продукта, либо, всего нашего it-продукта. А лит тайм — это весь период времени, от получения заказа от заказчика, до полностью получения от него уже оплаты, то есть полностью, весь период нашей работы.



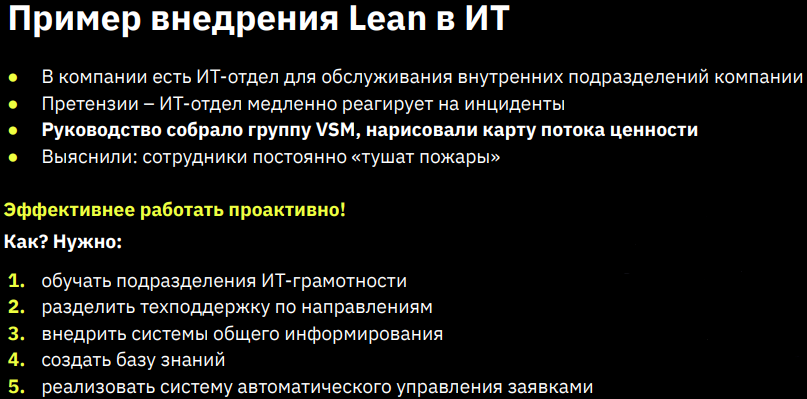
Можно ещё так вот, юмористически изобразить сайкл тайм и лит тайм. Для того чтобы создать, карту потоков создания ценностей, нам необходимо, собрать определённую группу людей,



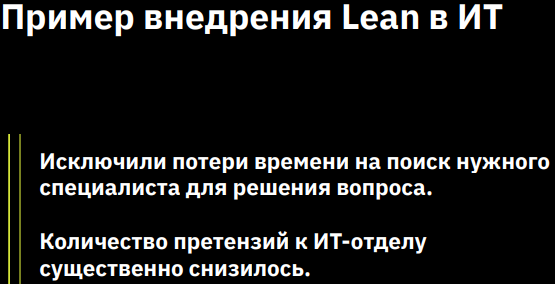
это так называемая группа VSM, обычно состоит от 6 до 10 человек, это люди, которые могут влиять на все потоки вашей компании, на те потоки, создания ценностей, которые вам нужны, при разработке продукта. Что это за люди? Эта группа может состоять из владельца продукта, различных управленцев, руководителей разработки, темлидов, менеджера по выпуску, то есть не важно, как называются эти роли в вашей компании, важно понимать, что в эту группу должны входить, именно те люди, которые могут, оказывать влияние, вносить какие-то изменения, в поток создания ценностей.



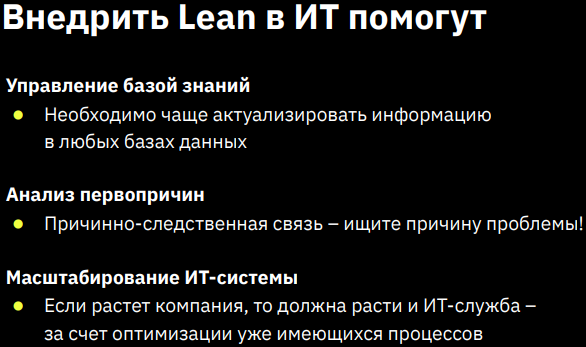
Давайте посмотрим на примере, как это выглядит. Допустим, мы нарисовали полностью, карту потока создания ценностей, что мы здесь видим, мы видим информационный и конечно же тайм лайн, если мы посмотрим, на информационный поток и на продуктовый поток, то увидим, что команда дизайнеров и команда разработчиков, получают исходную информацию по своим задачам, по продукту, из share point. A тестировщики и сборщики, из Excel, что здесь важно отметить, при передаче информации из share point в Excel, происходит определенная потеря данных, возможно, что-то никак, где-то возникают какие-то ошибки. Может произойти так, что тестировщики и сборщики, получат искаженную информацию, от наших дизайнеров и разработчиков, всего лишь из-за того, что у нас два разных канала, информационного потока. Адресовав все процессы, мы видим, что лучше объединить, либо выбрать share point либо Excel, и чтобы полностью все члены команды, получали данные по продукту из одного источника. Таким образом, мы избавимся от определенных потерь, из-за дефектов, и потерь по времени, ненужная транспортировка.



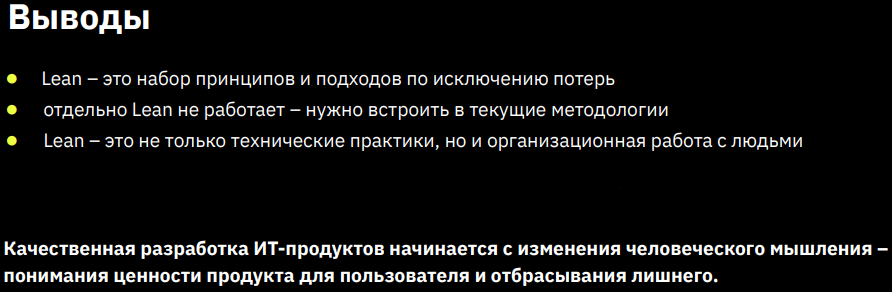
Давайте рассмотрим, ещё один интересный пример, внедрения лин в it-сферу, в it-компанию. Допустим, в компании есть it-отдел, который отвечает за обслуживание внутренних потребностей компании, то есть такая своеобразная, техническая поддержка, при этом очень много поступает жалоб, что команда не справляется со сроками отработки инцидентов, то есть возникают, какие-то инциденты, какие-то проблемы у работников компании. Они передают их в it-отдел, а те элементарно перегружены, не справляются со своими задачами. Руководство принимает решение, отрисовать карту потока создания ценности, ту самую карту VSM. Что у нас выясняется? Оказывается, работники занимаются постоянно тушением пожаров, то есть быстро, оперативно, стараются решить какие-то возникшие проблемы, с IT системами внутри компании, по которой им сигнализируют, соответственно работники этой компании. Но важно отметить, что эффективно работать про активно, то есть на опережение, не создавать эти инциденты. А как этого добиться, что для этого нужно сделать? Во-первых, обучить, специалистов различных подразделений компании, элементарной IT грамотности, элементарному управлению, какими-то IT системами. Значит нужно выделить специалиста, который будет проводить это обучение. В следующий момент, это разграничить техническую поддержку по направлению, чтобы можно было гораздо быстрее и легче найти специалиста, который сможет тебе помочь с твоей проблемой, внедрить системы общего информирования, какие-то массовые рассылки, о внедряемых новых изменениях ПО, которым пользуются определённые подразделения, создать базу знаний и реализовать систему автоматического управления заявками, чтобы система, автоматом распределяла заявки, между сотрудниками it-отдела, таким образом, систематизируем рутинную работу, внедряя в сети основные практики, мы во-первых, исключаем потери времени на поиск нужного специалиста, для решения какой-то задачи, и у нас снизилось, существенно количество претензий к it-отделу, потому что у нас, уже обученные специалисты. Люди понимают, знают, в каких программах они работают, их постоянно уведомляют о новых каких-то изменениях в данных программных продуктах, и в целом сократились it-инциденты, которыми должны заниматься соответственно люди, из нашей техподдержки.



Внедрить лин в IT практику, вам помогут следующие моменты, на которых, очень важно заострить внимание,



во-первых, это управление базой знаний, необходимо чаще актуализировать информацию, которая содержится у вас в различных базах знаний, будь то IT сфера, будь то производство или иная какая-то работа, которую вы делаете. Следующий момент, это анализ первопричин, всегда ищите ту самую причину, возникновения какой-то потери, потери времени, потери ресурсов при производстве продукта. Ищите причинно-следственную связь, это очень важно. Следующий момент, это масштабирование it-систем, конечно любые компании растут, расширяются, и по мере их роста, и расширения, растут IT система, базы знаний, вся инфраструктура, отделы, и так далее. Но важно отметить, что IT службу расширять необязательно, не нужно сразу же бежать и набирать, необходимое количество людей. Важно посмотреть на процессы, и расширить ИТ службу, за счёт оптимизации процессов и всей деятельности, что-то автоматизировать, где-то выявить потери, и устранить их, где-то убрать перегруз ваших сотрудников. Давайте, сделаем выводы и подытожим, всё вышесказанное,



во-первых, лин, это набор принципов и подходов, по исключению потерь, за счёт исключения потерь мы гораздо быстрее, можем получить наш готовый it-продукт, либо отдельные его элементы, и можно даже сказать, не только it-продукт, а вообще любой продукт, над чем вы работаете сейчас. Отдельно Лин не работает, данную концепцию важно встраивать в какие-то уже существующие методики, методологии и подходы на вашем производстве, либо в вашей работе, по созданию it-продукта. Лин, это не только технические практики, это больше организационная работа с людьми, то есть необходимо сформировать общее понимание, где мы теряем, где мы, наоборот, находим. Какие процессы можно усовершенствовать, улучшить и отрисовать естественно, ту самую карту процессов, карту потока создания ценностей. Увидеть, где вы теряете, какие есть проблемы, какие есть пробелы у вашей команды, и естественно, исключить это. Качественная разработка it-продукта начинается не с каких-то организационных моментов и выстраивания работы, а именно с человеческого мышления, важно донести до всех членов команды, над чем вы работаете, и в чём заключается, основная ценность вашего продукта, чтобы каждый член команды понимал это и с учетом этого подхода, уже проводил свою работу, разрабатывал отдельные элементы, рисовал дизайн, проводил вёрстку, и так далее.



Есть определённый набор шагов, чтобы внедрить лин в ваши it-проекты самый первый шаг, это зажгите андон. Андон, с японского языка — это своеобразная такая лампа, которую зажигают, с помощью определённого шнурка, дергают за шнурок и зажигается лампа, если мы говорим про лин. Что это значит, зажечь Андон? Это значит осветить проблему, вынести её на обсуждение, допустим если вы, при внедрении нового ПО, постоянно сталкиваетесь с одной и той же проблемой, важно остановить всю разработку, остановить всю работу, посмотреть, где возникает эта проблема, осветить её, и постараться её устранить естественно. Следующий второй шаг, это Генчи Генбуцу и Гемба, тоже с японского языка. Что это означает? В Японии есть такая практика, если на производстве что-то не работает, то собирают руководящий состав, собирают инженеров, ведут на линию производства, рисуют белым мелом круг, ставят туда этих людей, и они смотрят, наблюдают, за всем процессом. Что хотелось бы отметить, зачем это нужно? Чтобы устранить какой-то инцидент, который, очень достаточно трудно диагностировать, важно посмотреть, как происходит данная работа, как она делается на практике, как она делается в кабинетах, как она делается на производстве, или в целом как человек пишет код? Третий важный шаг Немаваси, с японского языка переводится как, подготовьте почву для изменений. Когда вы начинаете внедрять лин, хотите что-то изменить, исключить, определённые потери, изучить в целом весь поток создания ценности на вашем производстве, либо при разработке it продукта, вам важно всем сообща договориться, что вы меняете, и показать ценность этого изменения. Также важно, чтобы все члены команды, поддержали предложенную идею, только тогда вы сможете, действительно качественно, внедрить данную концепцию.

В итоге хотелось бы сказать, что, применяя все эти практики, определённые шаги, ту самую практику 5S, и все остальные элементы, про которые мы говорили, вы сможете удачно встроить данную лин концепцию, в концепцию бережного производства продукта, в работу на вашем предприятии, либо в разработку какого-нибудь it-продукта, в создание действительно практической ценности для вашего конечного пользователя, для вашего заказчика.