**Ручное рещение**

программа позволяет справиться с наростающим количеством слабо структурированной инофрмации по обслуживаемым программам. Количество програм, в которых принимаю участие должно увеличиться в дове, программы требуют ежеднеавной разработки и переключении с одной на другую, как можно быстрее. При этом программы разработаны многими людьми, и достаточно редко можно спросить о том как именно разрабатывать программу. Такая информация проедставляет одно из важных данных получаемых на митингах. Так же как и общее рекомендации, приходящие в письмах и чатах.

Изначально из меющегося опыта, было примененно ручное решение.

1. полезнчая ифформация записывается на митингах в блокнот, это быстно, но в последствии это сбало можно исопользовать, если не перенести в электронный вид

2. рекомендации сохраняются в один текстовый файл, и если это много строковыя рекомендация то можно применить разделитель ============ но зачастую вся рекомендация записанна просто однйо строкой

3. для каждой задачи создается отдельный текстовый файл, в котором подобно файлу рекомендаций ведутся пометки, также там могут присутствовать списки к примеру 1,2,3 и примеры кода

Выбор txt файлов не случаен, но ним просто можно поизводить поиск, через все документы и быстро находить нужный для детального разбора.

Так же ведется memory map для совсем новых задач и проектов, но это редко прикодит к накоплению полезной информации, больше порисовать, чтобы самому разобраться.

Вывод из ручного решения, оно хороше зарекомендовало себя, так как в отличии от запоминания, дает определенную гарантию, что можно бкдет вспомнить, то полезное, что вынес в какой то день, и вспомнить собранный опыт.

Пробема ручного решения, в том, что она не позволяет быстро создавать сценарии работы из уже имеющегося опыта, каждый раз, такой сценарий приходится все таки длолго искать и склеивать из той минимальной иформации, котораую удалось законспектировать.

**Checklist app**

это приложение как бы обьединяет ручной подход, в более стандартищированном виде.

А также ввобит понятие задания, и чеклист.

Каждое задание может обьединять в себя несколько чеклистов.

Чеклист состоит из строк, записсанных в тот или иной файл, будь то рекомендации или отдельная строка из файла, описывающего предыдущее задание. Главное, что это порядок действий, для решения той или иной задачи, проверки правильности решения, развертывания инфраструктуры.

Что инетересно в создании задачи, так то что в нее входят чек листы, похоже что это за частую линейные последовательности действий. Но само задание, это алгоритм, и он больше похож на дерево решения, граф из нескольких деревьев, возможно даже циклическое прохождение по списку мелких заданий из чеклиста.

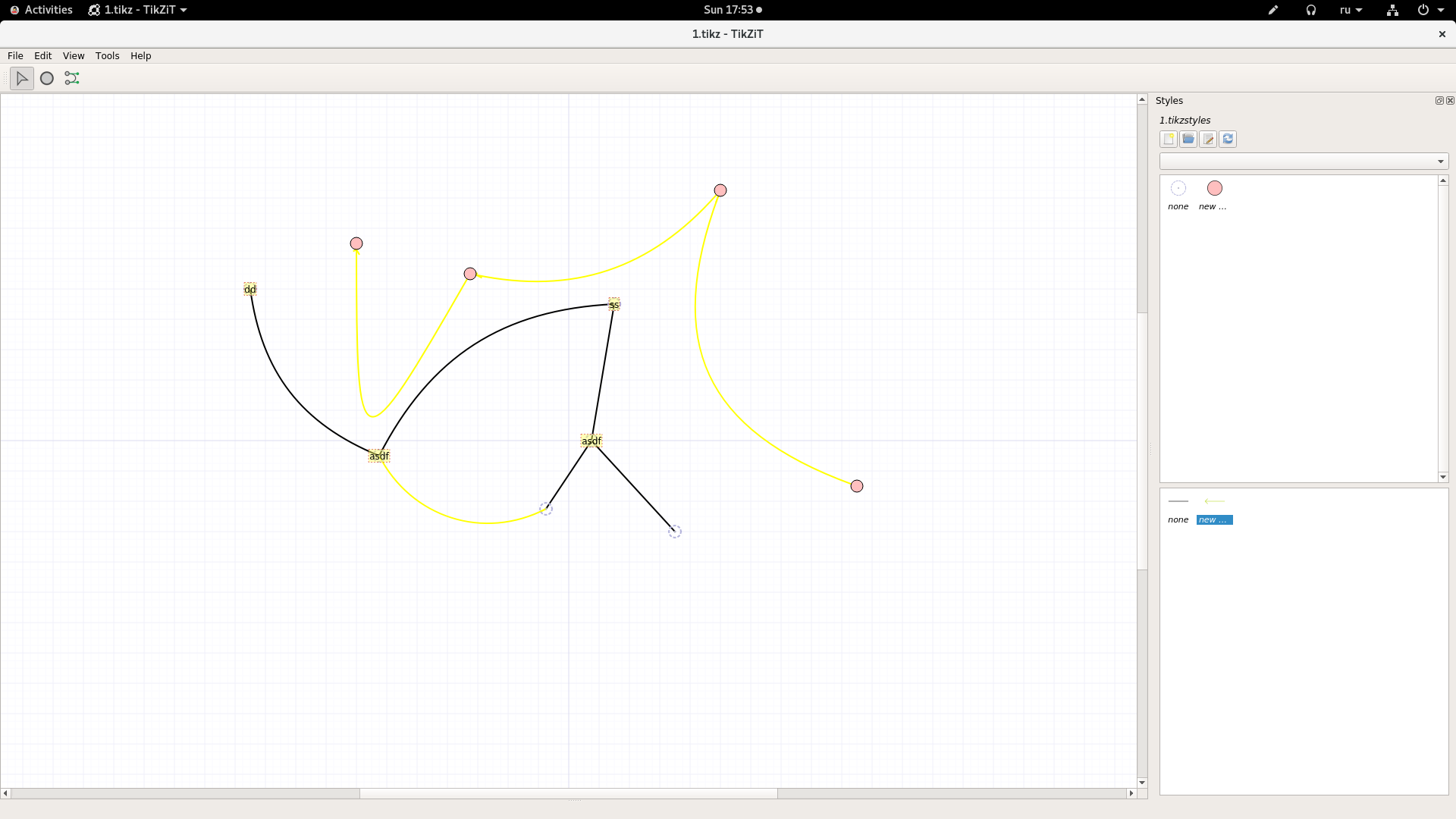
**Реализация**

Для того, чтобы быстро, возможно за пару рабочих дней, написать такой вот checklist app стоит выбрать консольное приложение.

Пусть оно будет позволять обрабатывать построчно файлы и вносить строки в чеклисты, создавать задания и прикреплять к ним чеклисты. Это достаточно простые действия которые в консоли можно делать последовательно.

Единственное, что нельзя сделать в консоли, так это выводить графы, деревья, выделять цветом, эту конечный план работы, с которым после быстрого анализа и приходитсыя работать.

Тут на много проше вызвать внешнее приложение, такое как TikZiT, примерно накидать для ее граф решения, и отдать его на ручное обновление. Выбранное приложение имеет простой тектовый формат файлов, который легко синхронизировать.



Как выглядит файл, прямо стили можно назвать понятно для программы: отложенно, сейчас, ненужно...

\begin{tikzpicture}

\begin{pgfonlayer}{nodelayer}

\node [style=none] (1) at (1.5, 4.5) {};

\node [style=none] (2) at (0.75, 0) {};

\node [style=none] (3) at (-10.5, 5) {dd};

\node [style=none] (4) at (1.5, 4.5) {ss};

\node [style=none] (5) at (3.5, -3) {};

\node [style=none] (6) at (0.75, 0) {asdf};

\node [style=none] (7) at (-6.25, -0.5) {asdf};

\node [style=none] (8) at (-0.75, -2.25) {};

\node [style=new style 0] (9) at (-7, 6.5) {};

\node [style=new style 0] (10) at (-3.25, 5.5) {};

\node [style=new style 0] (11) at (5, 8.25) {};

\node [style=new style 0] (12) at (9.5, -1.5) {};

\end{pgfonlayer}

\begin{pgfonlayer}{edgelayer}

\draw (3.center) to (7.center);

\draw (4.center) to (7.center);

\draw (4.center) to (6.center);

\draw (6.center) to (8.center);

\draw (6.center) to (5.center);

\draw [in=135, out=45, loop] (6.center) to ();

\draw [style=new edge style 0] (8.center) to (7.center);

\end{pgfonlayer}

Я например, расставляю как удобно для себя ноды, выделяю цветом те задания, которые нужно делать прямо сейчас, какие отложить, какие были лишними.

При выполненни задания, стоит продолжат вести файл с пометками.

Также интеренсно введение маркеров на всех уровнях, строки, чеклисты, задание, так как маркирование позводяет собирать данные, не зная точно слов, которые написанны.

Сквозной поиск по всем файлам внесенным в программу тоже кажется более удобным, чем через поисковик Notepad++. Дело в том, что часто в заданиях написанны имена классов, пути к проектам, которые можно сразу же подключать к поиску. Все таки это большая рутина, найти решение через пол года, когда уже все поменялось.

Вот я нарисовал в редакторе,Ю примерно пункты меню этой консольной программы:

Эта картинка, интерактивный граф, в нем можно отмечать сделанное, делать пометки, это как настольная игра, с ней разработка делается интереснее, более полно охватываешь состояние дел. Этого программа checklist app и долдна доиться.

