*Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования*

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет имени Н.Э. Баумана»******(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ РЛ

КАФЕДРА РЛ2

**Реферат на тему:**

**Разделение труда среди высококвалифицированных специалистов**

Студент Люзин М.А., группа РЛ2-101

Руководитель В.Н. Прокудин

Москва, 2020

Содержание

Введение 3

1. Причины разделения труда среди высококвалифицированных специалистов 4

2. Дрейфусовская модель приобретения навыков 5

2.1 Новичок 5

2.2 Продвинутый начинающий 5

2.3 Компетентный 6

2.4 Опытный 6

2.5 Эксперт 7

3. Модель junior-middle-senior 8

3.1 Junior-специалист 8

3.2 Middle-специалист 9

3.3 Senior-специалист 9

4. Модель Джозефа Сиджина 11

Заключение 14

Список литературы 15

Доклад по реферату 16

**Введение**

Разделение труда - исторически сложившийся процесс выполнения людьми своих специализированных видов деятельности в общем для всех деле, сопровождающийся обособлением, видоизменением, закреплением отдельных видов трудовой деятельности, который протекает в общественных формах дифференциации и осуществления разнообразных видов трудовой деятельности. Разделение труда как концепция появилась на свет благодаря творчеству Адама Смита в 1776 году в труде «Исследование о природе и причинах богатства народов»[1]. Можно сказать, что разделение труда среди высококвалифицированных специалистов способна увеличить производительность труда за счёт экономии времени на смене вида деятельности, более глубокого погружения работника в конкретный сектор работ, что даже может подтолкнуть к улучшению технологических процессов. На сегодняшний день разделение труда среди высококвалифицированных специалистов чаще всего происходит по Дрейфовской модели или по моделям Джозефа Сиджина или (junior-middle-senior)- с углублением в сферу IT.

1. **Причины разделения труда среди высококвалифицированных специалистов**

Любая современная компания, занимающаяся интеллектуальным трудом, вводит в том или ином роде деление сотрудников внутри каждого отдела на несколько различных групп.

Первой, и, навер ное, наиболее значимой причиной является упрощение понимания квалификации разработчиков. Так, главе отдела важно понимать, какому сотруднику необходимо поставить какую задачу для её успешного выполнения. Для этого можно воспользоваться объективной мерой оценки квалификации сотрудника. О системе оценок квалификации будет написано далее.

Вторая причина - определение зарплатных вилок. Например, во многих матричных организациях бывает так, что на одних и тех же позициях в соседних отделах сотрудники имеют одинаковый оклад. Это позволяет сотрудникам иметь большую внутриорганизационную мобильность, а также понижает вероятность недовольства сотрудников зарплатой.

Далее рассмотрим модели подобного деления сотрудников.

1. **Дрейфусовская модель приобретения навыков**

Данная модель была предложена сотрудниками центра Исследования Операций Университета Калифорния, братьями Стюарт и Губертом Дрейфусом в 1980 году [2].

Согласно этой модели, человек в своей работе проходит через пять различных этапов: новичок, продвинутый начинающий, компетентный, опытный, эксперт.

Рассмотрим каждый из этих уровней подробнее.

**2.1 Новичок**

Согласно исследованию, ожидается, что на этапе новичка работник не имеет опыта в работе. По этой причине для новичка крайне важно следовать прямым приказам, правилам и процедурам. Поскольку работа является для новичка чем-то новым, маловероятно, что такой сотрудник может быть креативным. Люди на этом этапе ещё не могут принимать собственные решения, и по этой причине они всё ещё изучают основы, выполняя приказы начальства. Наиболее важность частью этого этапа является то, что новичок должен получить полное понимания контекста информации, которая используется в этой специальности. Если рассматривать психологию такого сотрудника, то для них характерны следующие черты: новички сильно переживают за выполняемую работу, легко подвержены унынию в случае ошибочных результатов или результатов, не обговоренных в инструкции.

**2.2 Продвинутый начинающий**

На этом этапе модель приобретения навыков предполагает, что сотрудник достаточно практиковался с различными примерами работы или заданий. По этой причине сотруднику будет легче работать с различными сценариями, так как большинство из них уже основаны на его собственном опыте. Так как сотрудник уже имеет какой-то опыт, он увереннее подходит к решению задачи. Однако, если он столкнётся с какой-либо проблемой, велика вероятность того, что сотрудник будет винить не себя, а правила или процедуры, которых он придерживается. По этой причине у сотрудника может быть более низкое чувство ответственности, и, таким образом, на данной этапе он всё ещё нуждается в поддержке. Работники на такой ступени уже начинают формулировать для себя какие-то общие принципы, однако всё ещё не видят всей картины.

**2.3 Компетентный**

Достижение данной ступени зависит от рода деятельности сотрудника, но в среднем необходимый опыт в компании составляет порядка 3 лет. Ожидается, что компетентный сотрудник может работать эффективно, организованно и знает, что он делает, основываясь на своём предыдущем опыте. Важно, что такой работник имеет чувство ответственности за результат, и, следовательно, внимательнее подходит к решению проблемы. Подобные люди знают много правил и процедур, из-за чего им бывает трудно определиться с конкретным подходом к решению поставленной задачи, также они могут строить правильные модели проблемной области и эффективно ими пользоваться. Стоит заметить, что сотрудники на этой ступени развития способны устранять проблемы, с которыми они не сталкивались ранее, а также могут помогать новичкам и продвинутым начинающим. Типичная ошибка подобного работника - неправильно расставленные приоритеты при решении задач. Если сфера деятельности является технической, то, как правило, про таких людей говорят, что они «настоящие инженеры», так как их работа далеко не всегда следуюет какому-то шаблону.

**2.4 Опытный**

Человек достигает этой ступени, когда он хорош в какой-то определённой специальности. Опытный человек знает границы своих навыков, и может кретивно применять их в решении подчас очень сложных задач. Важно, что такой сотрудник видит критические аспекты ситуаций. Подобные сотрудники сделали большой прорыв в своей области, так как они постоянно оценивают выполненную работу и пересматривают свои подходы, чтобы в следующий раз быть наиболее эффективными. Такие сотрудники эффективно учатся на чужом опыте, а также всегда принимают во внимание контекст задачи, чтобы правильно расставлять приоритеты и быстро принимать решения.

**2.5 Эксперт**

Подобные сотрудники - основной источник знаний и информации в любой среде. Таким людям больше не нужны прямые правила, инструкции и процедуры для выполнения работы. Из-за большого опыта в конкретной области, эксперт неосознанно применяет соответствующие правила и указания к каждой конкретной ситуации. Как правило, у подобных сотрудников имеется большой ряд публикаций, в том числе и книг. Если вдруг возникает задача, которая является новой для эксперта, данная модель предполагает, что эксперт использует свои аналитические навыки для оценки ситуации. Важно, что эксперты используют личную интуицию для достижения наиболее значимых резальтутов. Стоит заметить, что эксперты - наиболее компетентные люди, а, как известно, подобные люди больше сомневаются в полноте своих знаний, чем малокомпетентные специалисты.

**2.6 Итоги модели Дрейфуса**

Для каждого сотрудника, независимо от типа работы или отрасли, крайне важно сперва следовать прямым правилам и рекомендациям, чтобы освоить задание. После того, как сотрудник становится более опытным, он становится менее зависимым от правил и инструкций, необходимых для выполнения работы или задачи. Также ожидается, что чем опытнее сотрудник, тем лучше он может справляться с неожиданными событиями. Модель Дрейфуса полезна для понимания уровня развития любого человека на любой работе. Важно понимать, что сравниваться должны люди в пределах одной специальности.

**3. Модель junior-middle-senior**

Данная модель прочно закрепилась в IT-сфере, так как многие задачи можно формализовать и чётко разбить между специалистами в каждом отделе компании. Говоря о таком разделении, многие компании берут в расчёт не только *hard skills* - непосредственно профессиональные умения сотрудника, но также и *soft skills* - коммуникационные, командные и другие «неспециальные» навыки человека [3]. В целом, эта модель тесно связана с Дрейфусовской моделью, однако выделяются иные уровни. Ошибочно судить о позиции человека по опыту в компании, то есть сказать, что для перехода со ступени на ступень требуется около трёх лет, по причине того, что не все люди имеют желания развивать свои навыки, или же из-за разной скорости обучения. Поэтому позиции привязаны к навыкам человека, его самостоятельности и уровень задач, которые человек может решить, как и говорилось ранее. Далее рассматрим краткую характеристику для каждого из них.

**3.1 Junior-специалист**

Путь к уровню *Junior* разработчика начинается, когда в компанию приходит вчерашний студент с опытом решения некоммерческих задач и так называемых pet projects – небольших проектов, реализованных рамках обучения. Когда он попадает в команду, на него обрушивается множество новой информации, но самое важное он получает от наставника. В этот момент человек должен демонстрировать быструю обучаемость, не задавать два раза одни и те же вопросы и постараться максимум времени уделить обучению и задачам. Обычно компании берут человека, если до уровня junior ему осталось не более 6 месяцев. Так происходит, потому что простых задач в разработке не так уж и много. Часто задачи приходится специально искать под такого специалиста, чтобы он мог на чем-то учиться. Важную роль для такого специалиста играет его наставник, и он напрямую влияет на скорость обучения.

Достижение уровня Junior означает, что человек перестал отнимать существенное время у наставника и начал самостоятельно и качественно решать задачи в единицы дней. Маркером может быть то, что наставник начал принимать его работу с первого раза. Если говорить про программистов, на этапе *junior* мелкие задачи – это чаще всего исправление некритичных ошибок, добавление и изменение элементов пользовательского интерфейса.

**3.2 Middle-специалист**

Работник на такой позиции - человек, который уже набрал определенный опыт. Он понимает процессы в команде, архитектуру продукта, продуктовый контекст и почему все так, а не иначе. Он уже не боится задач длиной в неделю и более. Может декомпозировать эту задачу, в том числе на задачи в единицы дней, которые можно отдать junior-специалистам. Маркеры того, что человек перешел в разряд middle: наставничество, предложения по архитектуре и процессу, конструктивное обсуждение плана спринта, идентификация плохих задач и, самое главное, успешное решение задач средней сложности длиной в неделю и более. Middle – это человек, компетенции которого признала команда. Такой человек может справиться с добавлением новой бизнес-логики в существующий сервис. Важно, что такой специалист владеет своей сферой ответственности, понимает контекст и способен принимать решения по реализации с учётом и технических нюансов, и с учётом взгляда со стороны заказчиков.

**3.3 Senior-специалист**

Это достаточно уверенный и опытный специалист, прошедший боевое крещение реальными проектами, переживший не один дедлайн и в одиночку успешно выбравшийся из-под обломков неожиданно рухнувшей системы. Основное отличие «старших» в том, что они уже полностью самостоятельны и могут не только единолично принимать правильные решения, но и нести за них ответственность перед заказчиками и пользователями разрабатываемых решений. Достаточно часто они управляют собственной командой менее толковых опытных сотрудников, при этом существенно меньше сами пишут код, зато могут беглым взглядом выявлять ошибки и упущения новобранцев, помогают им с расстановкой приоритетов и определением направления для развития. Кроме того, это уже однозначные эксперты в определенной области, которые высоко ценятся на рынке труда.

**4. Модель Джозефа Сиджина**

Говоря об IT-сфере, можно выделить замечательно структурированную работу Джозефа Сиджина (Sijin Joseph) [4], которая как раз объединяет различные качества сотрудника, как *soft skills*, так и *hard skills* в 4 категории.

В целом, данная модель подразумевает разбитие на 4 категории, первая обозначает новичка, последняя - самого высококлассного специалиста. Модель представлена в качестве таблицы, разбитой на следующие разделы: Computer Science (Информатика), Software Engineering (Программная инженерия), Programming (Программирование), Experience(Опыт), Knowledge (Знания).

В таблице 1 приведён пример такой таблицы. Стоит заметить, что, например, для причисления сотрудника к уровню 3 необходимо наличие уровней 1 и 2.

За каждую соответствующую реальному положению дел ячейку человек, проходящий этот тест, начисляет по одному баллу. Далее эти баллы суммируются по столбцам, затем суммы столбцов в зависимости от уровня подвергаются разным математическим операциям, затем результаты математических операций разных уровней складываются. В итоге использование такого математического подхода даёт значительный прирост на простых вещах, соответствующих уровню 1, и малый прирост, соответствующий уровню 4, что, в принципе, соответствует принципу 80-20.

|  |
| --- |
| Таблица 1 - пример матрицы Сиджина (переводённый на русский язык вариант из [5]) |
| Уровень | 1 | 2 | 3 | 4 |
|  | 2n | n2 | n | log(n) |
| Информатика - Computer Science |
| Структуры данных | Не понимает разницы между массивом и связным списком. | Может объяснить и использовать на практике массивы, связные списки, словари и т.д. | Понимает плюсы и минусы использования тех или иных базовых структур данных (размер памяти, время выполнения операций с данными, в чем разница между массивами и связными списками в этом плане). | Знание сложных структур данных, таких как B-дерево, Биномиальная куча и Фибоначчиевская куча, АВЛ-дерево, Красно-чёрное дерево, Косое дерево, Список с пропусками, TRIE-структуры и т.д. |

Так, человек, владеющий навыками первого, второго и третьего уровней для блока «Информатика» по теме «Структуры данных» может посчитать свой балл:

|  |  |
| --- | --- |
|  | (1) |

Возможно, данная модель является чересчур подробной, однако позволяет сравнить soft и hard skills сотрудников. В принципе, данная модель может использоваться также и для любой другой специальности, однако составить её может лишь специалист, имеющий большой опыт в данной сфере.

**Заключение**

Приведённые в реферате модели разделения труда среди высококвалифицированных специалистов помогают руководителям отделов и целых предприятий осуществлять эффективное распределение задач. Также данные метрики помогают специалистам объективнее оценить собственные умения согласно текущим нормам рынка и в перспективе указать на нужные направления в личностном развитии. Хоть большинство информации в данном реферате и касается сферы компьютерных технологий, основные понятия также могут быть переложены на любую другую сферу компетентными в этой сфере специалистами.

**Список литературы**

1. Аскильдсен Я. Э. Адам Смит и «невидимая рука» рыночного механизма // Теория и методы в социальных науках / Под ред. С. Ларсена; Пер. с англ. — М.: Московский государственный институт международных отношений (Университет); «Российская политическая энциклопедия» (РОССПЭН), 2004. C. 147—163.
2. Dreyfus, Stuart E.; Dreyfus, Hubert L. (February 1980). "A Five-Stage Model of the Mental Activities Involved in Directed Skill Acquisition" (PDF). Washington, DC: Storming Media. Retrieved June 13, 2010.
3. Отличия junior, middle и senior разработчиков — объясняют эксперты [Электронный ресурс] //tproger.ru. 2018. Дата обновления: 04.08.2018. URL: https://tproger.ru/experts/junior-middle-senior-developers-differences/ (дата обращения: 28.04.2020).
4. Programmer Competency Matrix [Электронный ресурс] //sijinjoseph.com. 2012. Дата обновления: н/д. URL: https://sijinjoseph.com/programmer-competency-matrix (дата обращения: 28.04.2020).
5. Матрица компетентности программиста, мой перевод. [Электронный ресурс] //https://github.com. 2018. Дата обновления: 25.11.2018. URL: https://github.com/omreps/programmer-competency-matrix (дата обращения: 28.04.2020).

**Доклад по реферату**

**Введение**

Разделение труда - исторически сложившийся процесс выполнения людьми своих специализированных видов деятельности в общем для всех деле, сопровождающийся обособлением, видоизменением, закреплением отдельных видов трудовой деятельности, который протекает в общественных формах дифференциации и осуществления разнообразных видов трудовой деятельности. Разделение труда как концепция появилась на свет благодаря творчеству Адама Смита в 1776 году в труде «Исследование о природе и причинах богатства народов»[1]. Можно сказать, что разделение труда среди высококвалифицированных специалистов способна увеличить производительность труда за счёт экономии времени на смене вида деятельности, более глубокого погружения работника в конкретный сектор работ, что даже может подтолкнуть к улучшению технологических процессов. На сегодняшний день разделение труда среди высококвалифицированных специалистов чаще всего происходит по Дрейфовской модели или по моделям Джозефа Сиджина или (junior-middle-senior)- с углублением в сферу IT.

**Причины разделения труда среди высококвалифицированных специалистов**

Любая современная компания, занимающаяся интеллектуальным трудом, вводит в том или ином роде деление сотрудников внутри каждого отдела на несколько различных групп.

Первой, и, наверное, наиболее значимой причиной является упрощение понимания квалификации разработчиков. Так, главе отдела важно понимать, какому сотруднику необходимо поставить какую задачу для её успешного выполнения. Для этого можно воспользоваться объективной мерой оценки квалификации сотрудника. О системе оценок квалификации будет написано далее.

Вторая причина - определение зарплатных вилок. Например, во многих матричных организациях бывает так, что на одних и тех же позициях в соседних отделах сотрудники имеют одинаковый оклад. Это позволяет сотрудникам иметь большую внутриорганизационную мобильность, а также понижает вероятность недовольства сотрудников зарплатой.

**Примеры разделения труда высококвалифицированных сотрудников**

Для наглядности стоит привести несколько примеров. Возьмём крупнейшую российскую IT-компанию - Яндекс. Как и в других подобных компаниях, Яндекс выделяет четыре категории специалистов: младший специалист, специалист, старший специалист, руководитель. Стоит заметить, что руководитель также принимает участие в проектах как ментор, который может помочь подчинённым советом, а также направить работу команды в нужное русло. Важно, что руководитель должен иметь более высокий уровень квалификации, чем старший специалист, о чём сказано на главном сайте компании. Яндекс набирает заинтересованных молодых людей вне зависимости от их возраста, ставя в приоритет их умения. Так, студент может прийти в команду младшим специалистом на стажировку, затем, если работа ангажирует его и он будет прилагать усилия, он может через какое-то время либо подняться на более высокую ступень в данной рабочей группе или же сменить отдел/специализацию внутри компании. Как это будет показано далее, HR-специалисты Яндекса могут просить руководителей оценивать кандидатов на должность по модели, схожей с моделью Джозефа Сиджина, по которой будут оценены и hard skills, и soft skills. Подобное разделение труда сотрудников помогает компании отойти от классической метрики «опыт работы в годах», которая, по сути, ничего не говорит о человеке, мотивируя тем самым сотрудников компании как можно усерднее стараться в решении каких-либо задач. Это, конечно же, влечёт за собой конкурентоспособность фирмы, а также позволяет фирме лучше реагировать на изменения на рынке.

**Дрейфусовская модель приобретения навыков**

Данная модель была предложена сотрудниками центра Исследования Операций Университета Калифорния, братьями Стюарт и Губертом Дрейфусом в 1980 году [2]. Согласно этой модели, человек в своей работе проходит через пять различных этапов: новичок, продвинутый начинающий, компетентный, опытный, эксперт. Для каждого сотрудника, независимо от типа работы или отрасли, крайне важно сперва следовать прямым правилам и рекомендациям, чтобы освоить задание. После того, как сотрудник становится более опытным, он становится менее зависимым от правил и инструкций, необходимых для выполнения работы или задачи. Также ожидается, что чем опытнее сотрудник, тем лучше он может справляться с неожиданными событиями. Модель Дрейфуса полезна для понимания уровня развития любого человека на любой работе. Важно понимать, что сравниваться должны люди в пределах одной специальности.

**Модель junior-middle-senior**

Данная модель прочно закрепилась в IT-сфере, так как многие задачи можно формализовать и чётко разбить между специалистами в каждом отделе компании. Говоря о таком разделении, многие компании берут в расчёт не только *hard skills* - непосредственно профессиональные умения сотрудника, но также и *soft skills* - коммуникационные, командные и другие «неспециальные» навыки человека [3]. В целом, эта модель тесно связана с Дрейфусовской моделью, однако выделяются иные уровни. Ошибочно судить о позиции человека по опыту в компании, то есть сказать, что для перехода со ступени на ступень требуется около трёх лет, по причине того, что не все люди имеют желания развивать свои навыки, или же из-за разной скорости обучения. Поэтому позиции привязаны к навыкам человека, его самостоятельности и уровень задач, которые человек может решить, как и говорилось ранее.

**Модель Джозефа Сиджина**

Говоря об IT-сфере, можно выделить замечательно структурированную работу Джозефа Сиджина (Sijin Joseph) [4], которая как раз объединяет различные качества сотрудника, как *soft skills*, так и *hard skills* в 4 категории.

В целом, данная модель подразумевает разбитие на 4 категории, первая обозначает новичка, последняя - самого высококлассного специалиста. Модель представлена в качестве таблицы, разбитой на следующие разделы: Computer Science (Информатика), Software Engineering (Программная инженерия), Programming (Программирование), Experience(Опыт), Knowledge (Знания).В таблице 1 приведён пример такой таблицы. Стоит заметить, что, например, для причисления сотрудника к уровню 3 необходимо наличие уровней 1 и 2.

За каждую соответствующую реальному положению дел ячейку человек, проходящий этот тест, начисляет по одному баллу. Далее эти баллы суммируются по столбцам, затем суммы столбцов в зависимости от уровня подвергаются разным математическим операциям, затем результаты математических операций разных уровней складываются. В итоге использование такого математического подхода даёт значительный прирост на простых вещах, соответствующих уровню 1, и малый прирост, соответствующий уровню 4, что, в принципе, соответствует принципу 80-20.

|  |
| --- |
| Таблица 1 - пример матрицы Сиджина (переводённый на русский язык вариант из [5]) |
| Уровень | 1 | 2 | 3 | 4 |
|  | 2n | n2 | n | log(n) |
| Информатика - Computer Science |
| Структуры данных | Не понимает разницы между массивом и связным списком. | Может объяснить и использовать на практике массивы, связные списки, словари и т.д. | Понимает плюсы и минусы использования тех или иных базовых структур данных (размер памяти, время выполнения операций с данными, в чем разница между массивами и связными списками в этом плане). | Знание сложных структур данных, таких как B-дерево, Биномиальная куча и Фибоначчиевская куча, АВЛ-дерево, Красно-чёрное дерево, Косое дерево, Список с пропусками, TRIE-структуры и т.д. |

**Заключение**

Таким образом, приведённые в реферате модели разделения труда среди высококвалифицированных специалистов помогают руководителям отделов и целых предприятий осуществлять эффективное распределение задач. Также данные метрики помогают специалистам объективнее оценить собственные умения согласно текущим нормам рынка и в перспективе указать на нужные направления в личностном развитии. Хоть большинство информации в данном реферате и касается сферы компьютерных технологий, основные понятия также могут быть переложены на любую другую сферу компетентными в этой сфере специалистами.