



Astropilot 29 ноября 2011 в 13:08

# Микроменеджмент: время создавать зомби

Управление проектами

Хабра приветствую всех, кому интересна эта тема.

Конечно же, микроменеджмент встречается не только в IT, но именно в этой сфере указанный черный ритуал может принести значительный вред процессу разработки и конечному результату, не говоря уже о профессиональном развитии сотрудников.

Наверное, для начала стоит дать определение этому катаклизму от управления. Итак, по мнению [businessdictionary.com](http://businessdictionary.com), микроменеджмент — это жесткий, тщательный, постоянной проводимый и часто демотивирующий контроль работы, выполняемой сотрудниками. Другими словами, если ваше начальство шагу не дает вам ступить, не проверив, правильно ли вы это сделали, постоянно контролирует все ваши действия и требует отчета даже о малейших движениях, — поздравляю, вы — потенциальная или уже вполне состоявшаяся жертва микроменеджмента.

## В чем же минусы микроменеджмента?

Самая большая опасность этого стиля управления — это превращение творческих, самостоятельно мыслящих, инициативных работников в безвольных зомби, исправно исполняющих поручения без блеска в глазах. Учитывая, что в IT процент людей, умеющих нестандартно мыслить и генерировать яркие идеи, очень высок, то такой стиль управления, как микроменеджмент, не только неразумен, но и в некоторых случаях способствует параличу проекта, конфликтам и даже уходу специалистов из команды.

## Чем еще опасен микроменеджмент?

Отсутствием доверия к команде со стороны ее руководителя и его запретом совершать ошибки, что приводит к остановке в развитии команды. Если люди чувствуют, что им не доверяют, то формирование команды замораживается. Все правильно, у зомби нет командного духа: их объединяет только географическая близость могил, из которых они восстали.

Микроменеджер думает, что знает, когда может доверять своей команде в принятии решений, а когда нет. Он следует одному принципу — разрешает своим людям действовать самостоятельно и свободно до тех пор, пока они действуют правильно в его понимании. Это что угодно, но никак не самостоятельность и не свобода. Свобода — это возможность действовать не так, как действовал бы на месте зомби заорганизованного разработчика сам ~~некромент~~ микроменеджер. В более широком смысле это тоже верно: право быть правым (в глазах микроменеджера) не имеет никакого отношения к свободе; настоящая свобода в том, чтобы иметь право быть неправым.

Постоянные проверки и слишком частый репортинг — еще одна отличная возможность убить доверие разработчика, его веру в свои силы и желание сотрудничать с руководителем команды. Ничто так не разрушает рабочий энтузиазм, как ощущение, что к тебе относятся как к недоразвитому ребенку, которого нужно поминутно контролировать. Помимо падения уровня морали, такая модель управления чревата неэффективным использованием времени самого менеджера. Получается, что над задачей разработчика трудится 2 специалиста: сам разработчик и менеджер, причем время последнего стоит гораздо дороже.

Очередной недостаток микроменеджмента — потеря из фокуса стратегического подхода к решению задач. Конечно, не такого стратегического подхода, как в анекдоте о мудром филине и мышах, которые хотели стать ежиками, а адекватного стратегического мышления при планировании процесса разработки. При потере фокусировки возникает опасность впасть в крайность тактического планирования, что сделает малоэффективным само руководство разработкой.

Еще один неочевидный враг менеджера — перфекционизм. Так как основное свойство перфекционизма — неконтролируемое распространение, то очень сложно бывает определить момент, когда перфекционизм переходит в микроменеджмент. Стремление к совершенству и вера в себя как в совершенство творит чудеса: микроменеджер видит себя всесильным боевым магом предпоследнего уровня, который держит руку на пульсе своего отряда прокачанных бойцов. Но наступает момент, когда брошенный вскользь взгляд, как в дурацких фильмах ужасов, вдруг выхватывает вместо привычных перекошенных героизмом лиц милые физиономии зомби.

Кроме того, разработчик, попавший в радиус действия микроменеджера на некоторое время, может не только остановиться в развитии, но и существенно испортить свою карму или даже карьеру. Сколько манны предстоит потратить на реабилитацию зомби и интеграцию его в другую команду? Разучиться самостоятельно думать легко, а на восстановление этого навыка уйдет драгоценное время и ресурсы.

## Неужели все так плохо и не бывает ситуаций, когда микроменеджмент допустим?

Конечно, бывает. Если, например, налицо проблемы с вовлеченностью команды в процесс. Скажем, массовый когнитивный диссонанс, сезонная депрессия или просто предпраздничное настроение, в результате чего команда просто выпадает из процесса и не хочет в него впадать по своей воле. Вот тогда самое время открывать свои некромантские конспекты и вычитывать оттуда хотя бы простенькое заклинание сотворения зомби и управления ими.

Иногда по каким-то личным причинам сотрудник вдруг начинает вести себя деструктивно по отношению к выполняемой работе или к коллегам. В этом случае микроменеджмент не повредит, а наоборот, поможет предотвратить неожиданные или неприятные события. Иными словами, если появляется угроза внутренней безопасности (проекта, информации, сотрудников), то микроменеджмент как стратегия поведения и защиты вполне оправдан.

Но оправданное применение микроменеджмента – это скорее исключение, чем правило.

В общем случае, лучше исключить эту черную магию из процесса разработки. Как это сделать?

- Вот несколько рекомендаций для начинающих и состоявшихся микроменеджеров:
- вспомните, что вы — **руководитель** команды, а не один из разработчиков или тестировщиков. Ваша роль в том, чтобы помогать вашим подчиненным применять их знания и опыт разработки, а не ваши собственные. В том, что вы прокачались до умопомрачительного уровня магии разработки ПО и так никто не сомневается, и не нужно лишний раз об этом напоминать. Не всякий мозг справится с постижением понятия бесконечности. Постарайтесь как можно быстрее и безопаснее для себя и окружающих осуществить свой переход в новое качество – от эксперта к руководителю.
  - сосредоточьтесь на **постановке задачи**, а не на объяснении способов ее решения. Сообщите ЧТО нужно сделать, а не КАК это нужно делать. Сообщите команде, что должно получиться в результате их совместных магических усилий, а также дайте необходимые пояснения по параметрам или ограничениям, которые нужно учитывать при выполнении задачи. Пусть команда сама обработает исходные данные, использует свой опыт и творческий потенциал и найдет-таки нужный способ действия. Не бойтесь делегировать полномочия, ведь с ними вы делегируете и ответственность, которая сама по себе является мотивацией.
  - объясните значимость того или иного этапа разработки **в контексте всего проекта**. Видение полной картины и взаимосвязей между этапами проекта помогает осознанно продвигаться вперед, систематизировать уже проделанную работу и прогнозировать предстоящие задачи.
  - задавайте команде больше вопросов и давайте **меньше готовых ответов**. Как руководитель задавайте вопросы всегда, когда это возможно, и давайте ответы только в случае реальной необходимости.
  - помните: **люди, а не ресурс! (с)** Разработчиков можно воспринимать как цифры и человеко-часы при эстимации проекта, но когда менеджер работает с командой, лучше строить партнерские отношения с командой, так как цели и у разработчиков, и у руководителя проекта общие.

Постарайтесь избежать использования ~~некромантских практик~~ микроменеджмента, ведь кроме переконтроллированных зомби есть еще столько боеспособных и эффективных юнитов!

Очень хотелось бы услышать от уважаемых хабраграждан, как бороться с микроменеджментом в полевых условиях силами разработчиков. Особенно интересуют случаи успешной борьбы.

Теги: [менеджмент](#), [микроменеджмент](#), [разработка](#)

↑

+35

↓

🔖


76

👁

10,7k

💬

17



90,0

Карма

0,0

Рейтинг


5

Подписчики

Irina @Astropilot

IT marketer

## Комментарии 17



eyeofhell

29 ноября 2011 в 14:37

🔖

🔖

↑

+11

↓

Очень хорошо написано. Всячески поддерживаю. Тоже не люблю микроменеджмент. Но что могу сказать по последним пяти годам руководства разработкой (до этого более десяти лет работал программистом / тим лидом, в целом программирую неплохо):

https://habr.com/ru/post/133601/

2/5

— сосредоточьтесь на постановке задачи, а не на объяснении способов ее решения. Сообщите ЧТО нужно сделать, а не КАК это нужно делать. Сообщите команде, что должно получиться в результате их совместных магических усилий, а также дайте необходимые пояснения по параметрам или ограничениям, которые нужно учитывать при выполнении задачи. Пусть команда сама обработает исходные данные, использует свой опыт и творческий потенциал и найдет-таки нужный способ действия.

Как показывает практика — не найдут :( То есть они все сделают, и ЭТО даже будет удовлетворять входящим требованиям, но находящийся внутри код... Он... Как бы так помягче сказать... Будет несколько запутан и не очень готов к дальнейшим изменениям, даже если нарисовать goadmap на коды вперед. И рано или поздно руководитель разработкой ПО, который только ставит задачи и проверяет внешние требования и ограничения, столкнется с классической ситуацией, когда каждое следующее изменение программного продукта будет обходиться все дороже. А в какой-то момент времени будет дешевле переписать все с нуля :(

И это не потому что разработчики плохие. Просто как-то так на практике получается, что в ряде задач, если нужно сделать что-то более сложное и долгоживущее нежели одноразовый сайт или одноразовая автоматизация бизнеса, требуется какой-то чудовищный опыт чтобы код через полгода оставался поддерживаемым :( найти на рынке команду таких разработчиков — трудно и дорого. Приходится работать со средним уровнем по больнице.

В результате — как с фотошопом, который пять раз переписывали с нуля (или сколько там).

И что самое неприятное — я до сих пор до конца не понял что с этим делать. То есть есть наработки, идеи, я их потихонечку пробую на людях и процессах — но серебряной пули отлить не получается :( Как-то и с микроменеджментом плохо, и без микроменеджмента тоже не все гладко. А постоянный code review всего кода делать если разработчиков больше пяти человек уже и не получается O\_O.

Вот такие вот пироги, с глазами. Их едят, они глядят :).



**Etherial** 29 ноября 2011 в 18:14

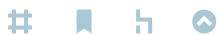


А с этим ничего не поделать. Любые хорошие продукты переписываются по несколько раз.

В любом случае у каждого человека свой взгляд на то, как надо писать код и просто должны быть регламенты, где есть основные договоренности, а все остальное — простор для творчества. Иначе получается вам нужны не специалисты, а дополнительные манипуляторы для себя, чтобы код быстрее писать можно было :)



**Etherial** 29 ноября 2011 в 18:17



Ну и конечно забыл написать, что любой код должен рефакториться периодически, чтобы оттянуть тот момент, когда придется выбросить весь код полностью и начать заново :)



**orcy** 29 ноября 2011 в 19:22



> а все остальное — простор для творчества

Когда с этим творчеством начнут вылазить многочисленные проблемы любой тимлид/менеджер пожалеет что оставил это на самотек. Однажды предоставив разработчикам такую свободу у вас накопится большое количество проблемного кода и ближе к релизу вы не будете знать что с этим делать. Решение довольно простое: если вы не уверены в качестве кода кого-то из разработчиков (например новый член команды), делайте постоянные code review. Review делает не обязательно менеджер, так что это вообще говоря не менеджмент. Также review это один из немногих способов junior разработчикам повысить качество кодинга.



**Colwin** 30 ноября 2011 в 12:05



«Совершенный код» и «Чистый код» сильно правят мозги :-)

Особенно переполненные ООП'ом или функциональным стилем, синтаксическим сахаром и т.д.

Мое личное видение: код может быть не лучшего вида, но высокая связность и низкая связанность должны сохраняться, иначе код потом трудно будет переписать.

Если же эти два критерия выполнены — можно скрепя сердце оставить низкоуровневые переплетенные конструкции в их локальном ящике, закрыть его интерфейсом, специфицировать — и пусть работает (если, конечно, работает). До тех пор, пока не понадобится его расширить — тогда и перепишем, если текущий вариант трещит по швам.

Code review за своей командой, ИМНО, проводить нужно, но не тыкать на не критичные вещи. Если такие находятся — лучше отдайте этот кусок на review другому коллеге, пусть он их найдет. А вот критические вещи все равно нужно контролировать кому-то достаточно опытному. Особенно когда дело касается оптимизации запросом путем создания индексов, когда ищутся пути избавлений от deadlock'ов и т.д.



**Mavim** 29 ноября 2011 в 19:28



да, кстати, интересная идея иногда просачивается, она такая ненавязчивая, но решает много проблем:

«клонирование себя» ) но когда реально себе это представляешь, то может прийти мысль, что не сработаемся )

> люди, а не ресурс!

Согласен, видимо Вам везет на инициативных.

На мой взгляд, инициатива — это одно, а разработка приложения с продуманным будущим — это другое.

Я не против инициативы, я против быдлокода.

Закрывать глаза на отклонения можно до поры до времени, потом Вам придется эту часть переписать.  
Микроменеджмент, он больше от безвыходности, когда вроде бы и рад им не заниматься, но как ни глянешь, лучше бы не смотрел и думал, что все в порядке.

Не знаю как с этим бороться, надо просто заниматься наставничеством, люди разные, перспективных хватает, многие боятся делать по-своему, таким достаточно дать волю и немного подправлять. Есть разработчики, не осознающие что они делают, с ними сложнее, тогда и возникает этот ступор и желание давать только маленькие задачи, пока человек не раскроется, показывать не только направление, но и «куда нажимать»

 **Astropilot** 29 ноября 2011 в 21:55

# 📌 ↻

↑ +1 ↓

Клонирование себя — любопытная идея, главное, чтобы она не стала общедоступной :)  
Я думаю, что микроменеджмент не только от безвыходности, а еще оттого, что управление — это скорее искусство, а не технология. Не хватает метрик, на основании которых можно было бы сколько там нужно раз мысленно щелкнуть хвостом, чтобы все было в порядке. Поэтому приходится постепенно набивать опыт и приходить к успеху со своим личным паноптикумом убитых или не взлетевших проектов.  
Наставничество — это хорошо, только к этому тоже не каждый способен. И, наверное, чем меньше руководитель к этому способен, тем чаще ему приходится прибегать к микроменеджменту.

 **DizelGenerator** 29 ноября 2011 в 21:08

# 📌

↑ +2 ↓

Автор, а вы сам — разработчик или менеджер? Даете правильные советы менеджерам — наверное, менеджер, однако, в контексте явно звучит недовольство микроменеджментом, примененным к вам же — наверное, разработчик.

Постоянное довление (от «довлеть») над душой никогда не будет результатом качественной работы, это раз.

Два — архитектура проекта все-таки может задаваться опытным специалистом, а накодить код могут и менее опытные, таким образом, остается следить за архитектурой, а это может быть один человек из команды программистов, с которым менеджер может общаться более неформально и на равных, что, мне кажется, способствует качеству и препятствует скорой кончине одной версии проекта с заменой ее на другую с нуля.

И три — зачастую есть не так много времени, чтобы выдумать нереально крутую архитектуру, которая останется на века. Нет времени и все тут, и тут приходится выбирать вариант «давайте сделаем менее мудрено и правильно, чтоб Фаулеру не стыдно было показать, но более быстро», что в достаточно опытных руках, опять же, не приводит к фатальным результатам.

 **Astropilot** 29 ноября 2011 в 23:39

# 📌 ↻

↑ +1 ↓

Большинство менеджеров так или иначе посидело в свое время на стульчиках разработчиков и тимлидов, почитало интересных книжек, посетило n-ное количество тренингов и конф и поумнело от всего вышеперечисленного.  
Аллергия на микроменеджмент — это здоровая реакция любого организма, который когда-либо испытывал его на себе (в любой профессиональной сфере), тут главное выработать иммунитет.

 **Colwin** 30 ноября 2011 в 12:07

# 📌 ↻

↑ +1 ↓

На любую силу найдется другая сила.  
На менеджеров тоже есть кому надавить сверху...

НЛО прилетело и опубликовало эту надпись здесь

 **png** 6 декабря 2011 в 21:27

# 📌

↑ +1 ↓

Статья очень понравилась. Большое спасибо автору.  
В ответ на:

«Очень хотелось бы услышать от уважаемых хабраграждан, как бороться с микроменеджментом в полевых условиях силами разработчиков. Особенно интересуют случаи успешной борьбы.»

Конструктивный диалог и предложение более качественных решений позволяют побороть это явление.  
«управляющий» (если он адекватен) видит, что человек в теме, и дает подчиненному свободу решать задачу по его схеме.  
Побывал в обоих ролях. Мне удавалось доказывать, свое мнение. И мне тоже доказывали мнение те, кем мне доводилось руководить.  
Так происходит потому, что человек решающий задачу на месте в самой задаче ориентируется лучше(если, конечно, он не начинающий, или просто новый человек в проекте, или ещё хуже — быдлокодер), чем руководитель.

А теперь позволю себе оставить свое мнение по поводу микроменеджмента:  
я думаю, что он необходим только в одном случае — как одно из средств для обучения.

Я делал так:  
сначала давал повозиться человеку в своем коде и документации 1-2 дня; пусть сам разберется;  
если не разобрался, давал ссылки на документацию, литературу и на примеры — как нужно.




если и это не помогло, то корректировал решение сам, либо часть решения, объяснял как нужно и почему так, заставлял доделать — как нужно; возможно на это уходило несколько итераций

Далее следующая задача по той же схеме. После нескольких задач человек растет быстрее, и достаточно просто проглядывать его код на явные косяки.

Пару раз схема не срабатывала. В проекте было много новичков, не хватало времени работать с каждым. Да и объясняю я не очень понятно. В итоге, быдлокод был жутким... Но к концу проекта народ стал писать по лучше.


Итог, мы все учились понемногу...)

 **Astropilot** 6 декабря 2011 в 22:35

# 📌 ↻ ⬆

↑ 0 ↓

Спасибо!  
По поводу применения микроменеджмента в обучении, соглашусь, это полезная практика. В большинстве случаев приносит результат, правда, необходим контроль, а на это уходит время. Однако, если в планах долгосрочная работа с командой, то есть смысл ее растить и вкладывать в это время и другие ресурсы.

 **png** 6 декабря 2011 в 22:43

# 📌 ↻ ⬆

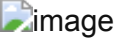
↑ 0 ↓


А куда деваться. Если дали человека. Нужно с ним так или иначе работать.  
А менеджеры при этом думают, что работа при этом пойдет быстрее. как же...

 **ded\_Sergei** 10 января 2012 в 07:37

# 📌

↑ +1 ↓

Написано хорошо. Внесу свои 5 копеек.  
Выбор микроменеджмента это должен быть осознанный выбор руководителя. Существует такая теория как ситуационная теория лидерства в частности теория жизненного цикла Херси и Бланшара.По ней стиль руководства и лидерства выбирается от “зрелости” исполнителей. Есть даже диаграммка по этому поводу.  
  
«Зрелость не следует определять в категории возраста. Зрелость отдельных лиц и групп подразумеваем способность нести ответственность за свое поведение, желание достигнуть поставленной цели, а также образование и опыт в отношении конкретной задачи, которую необходимо выполнить.»

 **Astropilot** 19 января 2012 в 17:57

# 📌 ↻ ⬆

↑ 0 ↓

Огромное спасибо за «5 копеек»! Многое стало на свои места в этой теме, по крайней мере для меня. Учитывая эту модель, микроменеджмент — просто один и паттернов управления, который нужно применять уместно и дозировано.

 **rednaxi** 10 января 2012 в 08:05

# 📌

↑ 0 ↓

про Бланшара тут уже написали, от себя хочу порекомендовать книгу «менеджер за одну минуту». Срубает на корню всю идею микроменеджмента.