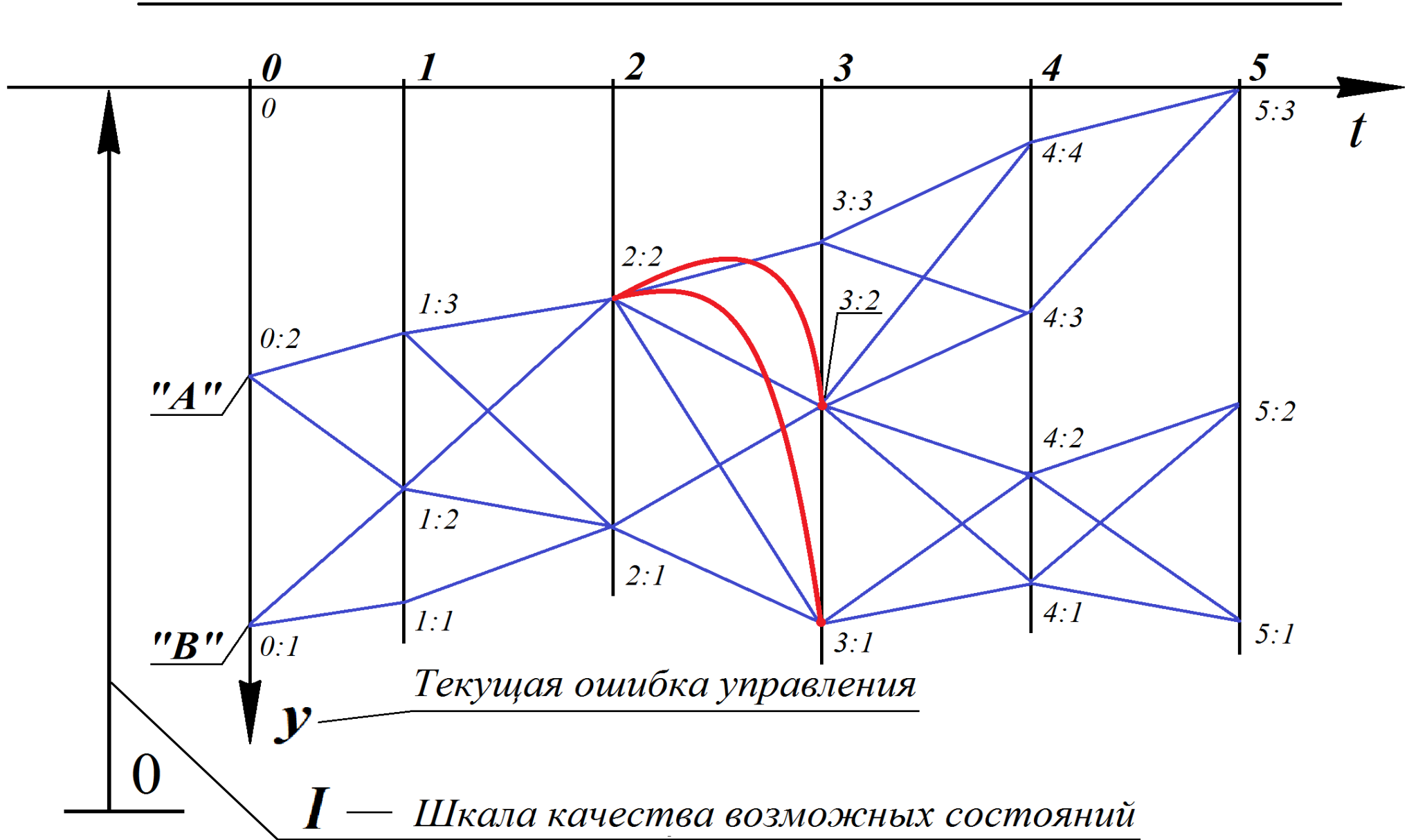


Метод динамического программирования

Множество возможных состояний



Вектор целей

1. Хочу ...
2. Хочу ...
3. Хочу ...

КОНЦЕПЦИЯ (общий замысел)

Сценарий и средства достижения 1 цели

Сценарий и средства достижения 2 цели

Сценарий и средства достижения 3 цели

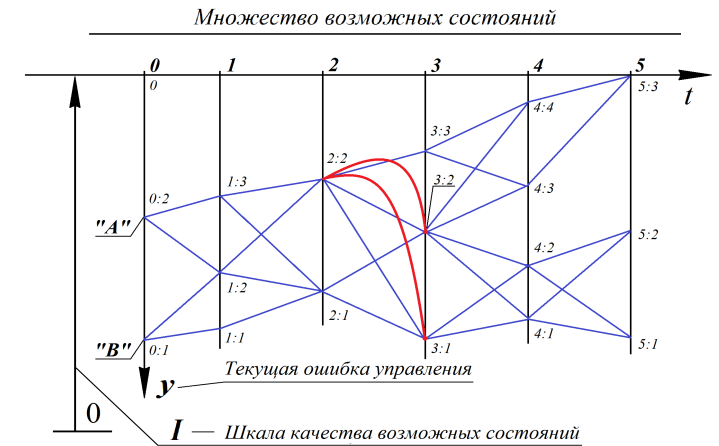
Вектор текущего состояния

1. Имеет место ...
2. Имеет место ...
3. Имеет место ...



У Валеры есть **МЕЧТА**, но у него смутные представления о том, как он будет её осуществлять...

Валера



Валера прошёл курс по Теории управления и узнал о МЕТОДЕ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ, который поможет ему сформировать КОНЦЕПЦИЮ осуществления его МЕЧТЫ.

Метод динамического программирования работоспособен, если формальная интерпретация реальной задачи позволяет выполнить следующие условия:

1. Рассматриваемая задача может быть представлена как N -шаговый процесс, описываемый соотношением:

$X_{n+1} = f(X_n, U_n, n)$, где n — номер одного из множества возможных состояний системы, в которое она переходит по завершении n -ного шага; X_n — вектор состояния системы, принадлежащий упомянутому n -ному множеству; U_n — управление, выработанное на шаге n (шаговое управление), переводящее систему из возможного её состояния в n -ном множестве в одно из состояний $(n + 1)$ -го множества. Чтобы это представить наглядно, следует обратиться к рис. 4, о котором речь пойдет далее.

2. Структура задачи не должна изменяться при изменении расчетного количества шагов N .

3. Размерность пространства параметров, которыми описывается состояние системы, не должна изменяться в зависимости от количества шагов N .

4. Выбор управления на любом из шагов не должен отрицать выбора управления на предыдущих шагах. Иными словами, оптимальный выбор управления в любом из возможных состояний должен определяться параметрами рассматриваемого состояния, а не параметрами процесса, в ходе которого система пришла в рассматриваемое состояние.

Чисто формально, если одному состоянию соответствуют разные предыстории его возникновения, влияющие на последующий выбор оптимального управления, то метод позволяет включить описания предысторий в вектор состояния, что ведёт к увеличению размерности вектора состояния системы. После этой операции то, что до неё описывалось как одно состояние, становится множеством состояний, отличающихся одно от других компонентами вектора состояния, описывающими предысторию процесса.

5. Критерий оптимального выбора последовательности шаговых управлений U_n и соответствующей траектории в пространстве формальных параметров имеет вид:

$$V = V_0(X_0, U_0) + V_1(X_1, U_1) + \dots + V_{N-1}(X_{N-1}, U_{N-1}) + V_N(X_N).$$

Критерий V принято называть *полным выигрышем*, а входящие в него слагаемые — *шаговыми выигрышами*. В задаче требуется найти *последовательность шаговых управлений* U_n и траекторию, которым соответствует максимальный из возможных *полных выигрышей*. По своему существу полный “выигрыш” V — мера качества управления *процессом в целом*. Шаговые выигрыши, хотя и входят в меру качества управления процессом в целом, но *в общем случае* не являются мерами качества управления на соответствующих им шагах, поскольку метод предназначен для оптимизации управления процессом в целом, а *эффективные шаговые управления* с большим шаговым выигрышем, но лежащие вне *оптимальной траектории*, интереса не представляют. Структура метода не запрещает при необходимости на каждом шаге употреблять критерий определения шагового выигрыша V_n , отличный от критериев, принятых на других шагах.

С индексом n — указателем-определителем множеств возможных векторов состояния — в реальных задачах может быть связан некий изменяющийся параметр, например: *время, пройденный путь, уровень мощности, мера расходования некоего ресурса* и т.п. То есть метод применим не только для оптимизации управления процессами, длящимися во времени, но и к задачам оптимизации многовариантного одномоментного или нечувствительного ко *времени* решения, если такого рода “безвременные”, “непроцессные” задачи допускают их многошаговую интерпретацию.

Правила использования МДП:

Шаг 3А Шагом



1. Достижение цели (желаемого состояния) рассматривается как **ПРОЦЕСС В НЕСКОЛЬКО ШАГОВ** (**ПРОЦЕСС УПРАВЛЕНИЯ ДЕЛИМ на ШАГИ: каждый шаг – приближает к цели**). Можно описать будущую **ПОБЕДУ**, и двигаться к **НАЧАЛУ** (к настоящему) описывая состояния перед ней.
2. **СТРУКТУРА ЗАДАЧИ НЕ ДОЛЖНА ИЗМЕНЯТЬСЯ** при изменении расчётного количества шагов (**ЖЕЛАЕМАЯ МЕЧТА – НЕДОЛЖНА ПЕРЕСМАТРИВАТЬСЯ: РЕШИЛ и ВСЁ!**).
3. **ОПИСАНИЕ СОСТОЯНИЯ СИСТЕМЫ НЕ ДОЛЖНО ИЗМЕНЯТЬСЯ** от количества шагов (**ТВОЁ ОТНОШЕНИЕ К СЕБЕ и к ДЕЛУ НЕДОЛЖНО МЕНЯТЬСЯ, КАК БЫ НИ ДАЛЕКА и НЕ ТРУДНА БЫЛА ЦЕЛЬ – «Я и ДЕЛО - НЕПОКОЛЕБИМЫ»**). **НЕ ПРИНИЖАТЬ СЕБЯ и ДЕЛО!**
4. **НЕЛЬЗЯ ОТРИЦАТЬ ВЫБОР УПРАВЛЕНИЯ НА ПРЕДЫДУЩИХ ШАГАХ.** Следующий шаг должен определяться достигнутым состоянием, а не впечатлениями о «неудачах» сделанных шагов. **НЕЛЬЗЯ СКОРБИТЬ О ПРОШЛОМ.** *Не важно где ты находишься, важно то, к чему ты стремишься (ВАЖНЫ ТВОИ НАМЕРЕНИЯ).* **НЕУДАЧА – НЕОБХОДИМАЯ ЧАСТЬ ОБРЕТЕНИЯ ПОЛНОЙ ПОБЕДЫ.**

**НЕ ПРЕДАВАЙ
МЕЧТУ!**

**НЕВОЗМОЖНО
ПОБЕДИТЬ
ТОГО - КТО НЕ
СДАЁТСЯ.**

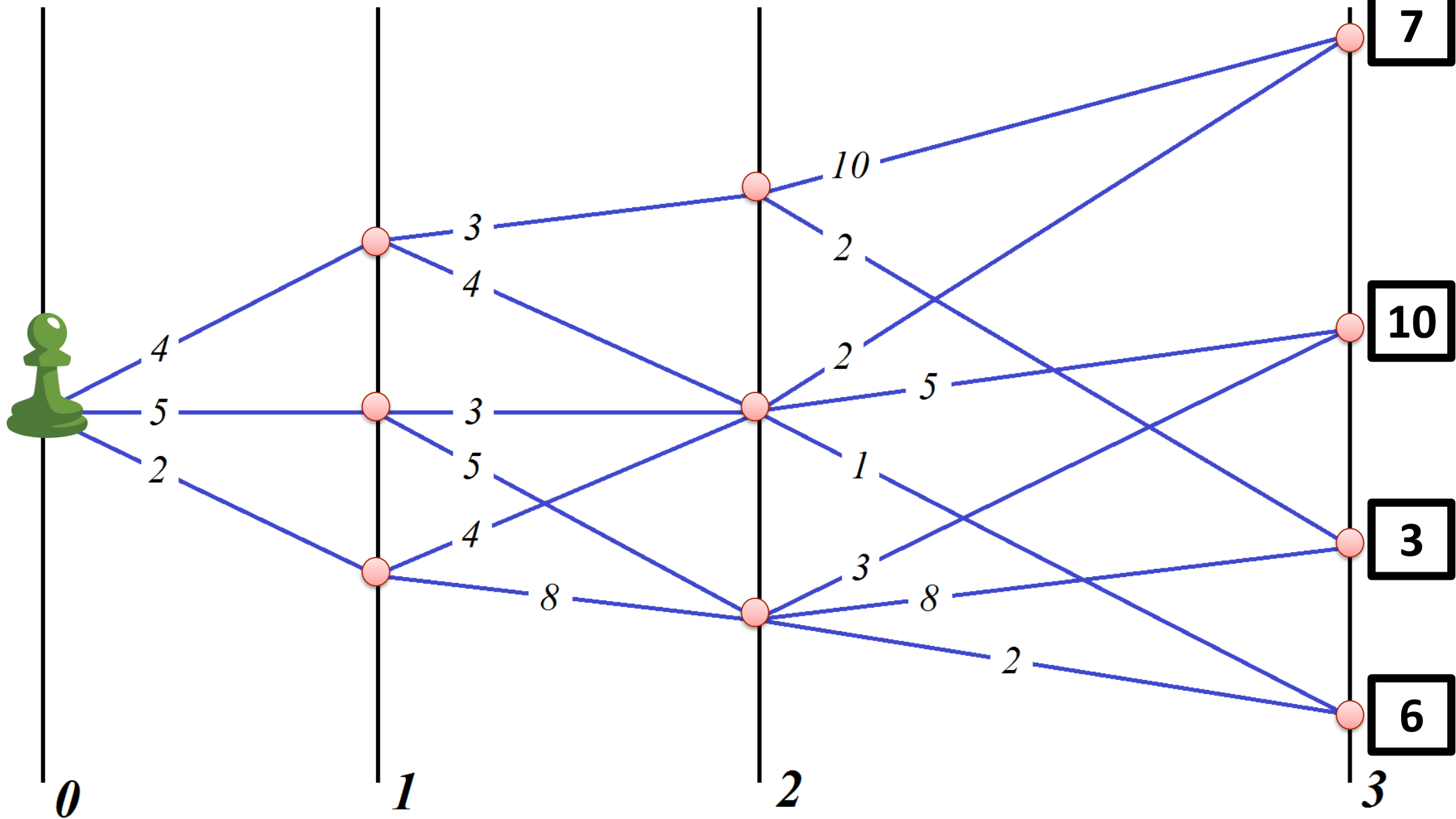


**Успех – это когда 9 раз
упал, но 10 раз поднялся**



Шаговые выигрыши

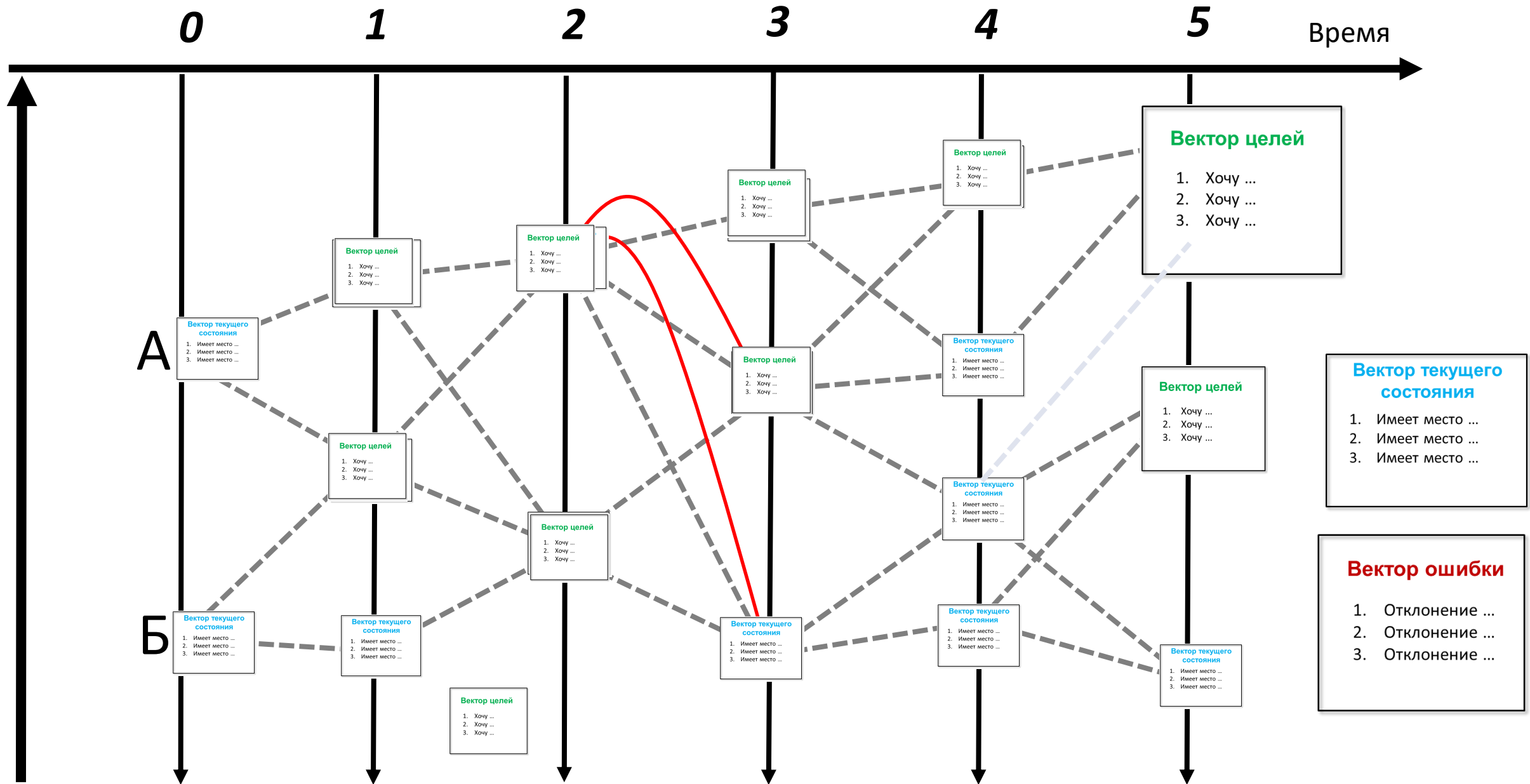
Полные выигрыши



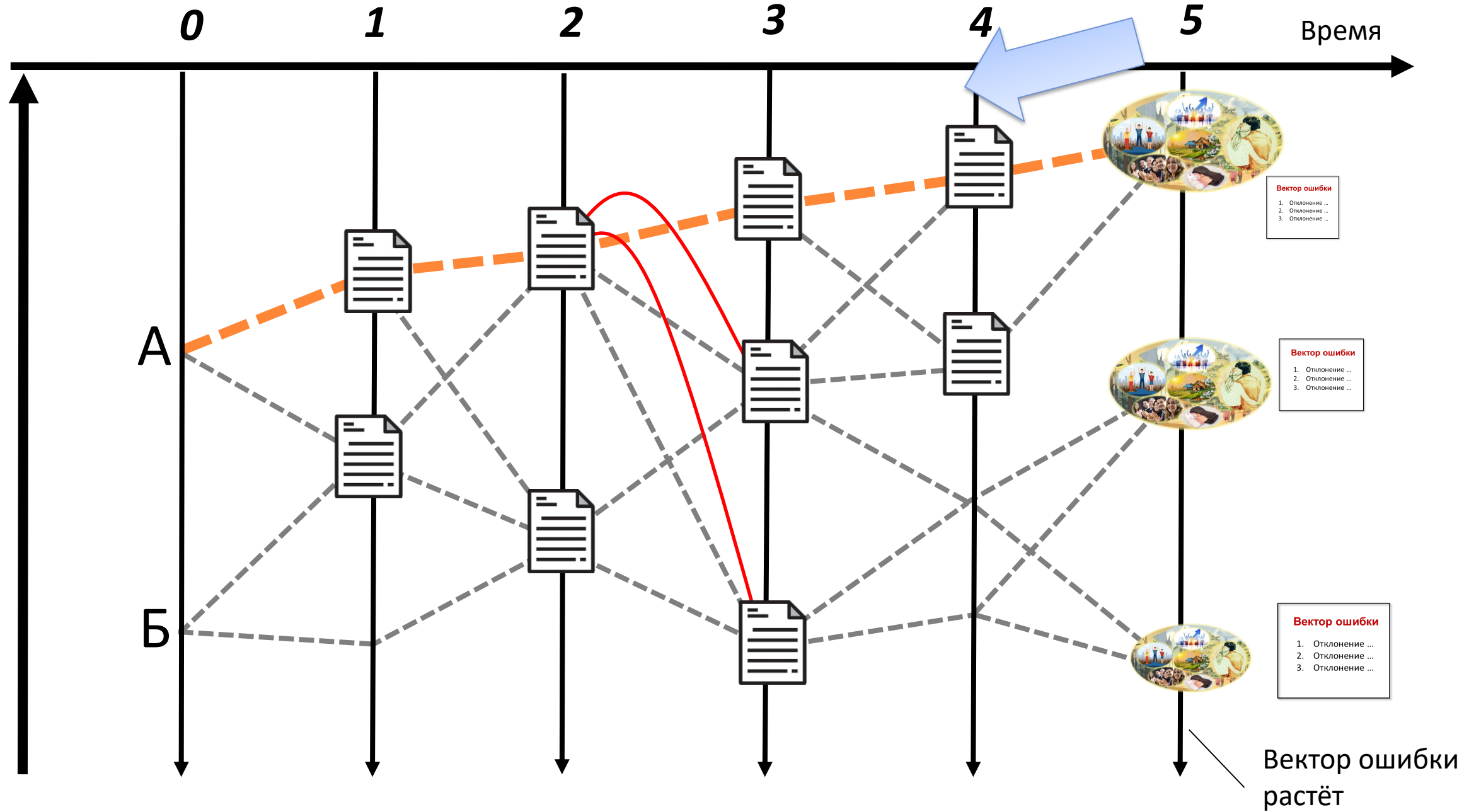
4+3
4+4
5+5
2+4

Множество состояний

Множество возможных состояний



Множество возможных состояний



Матрица возможных состояний (судьба)

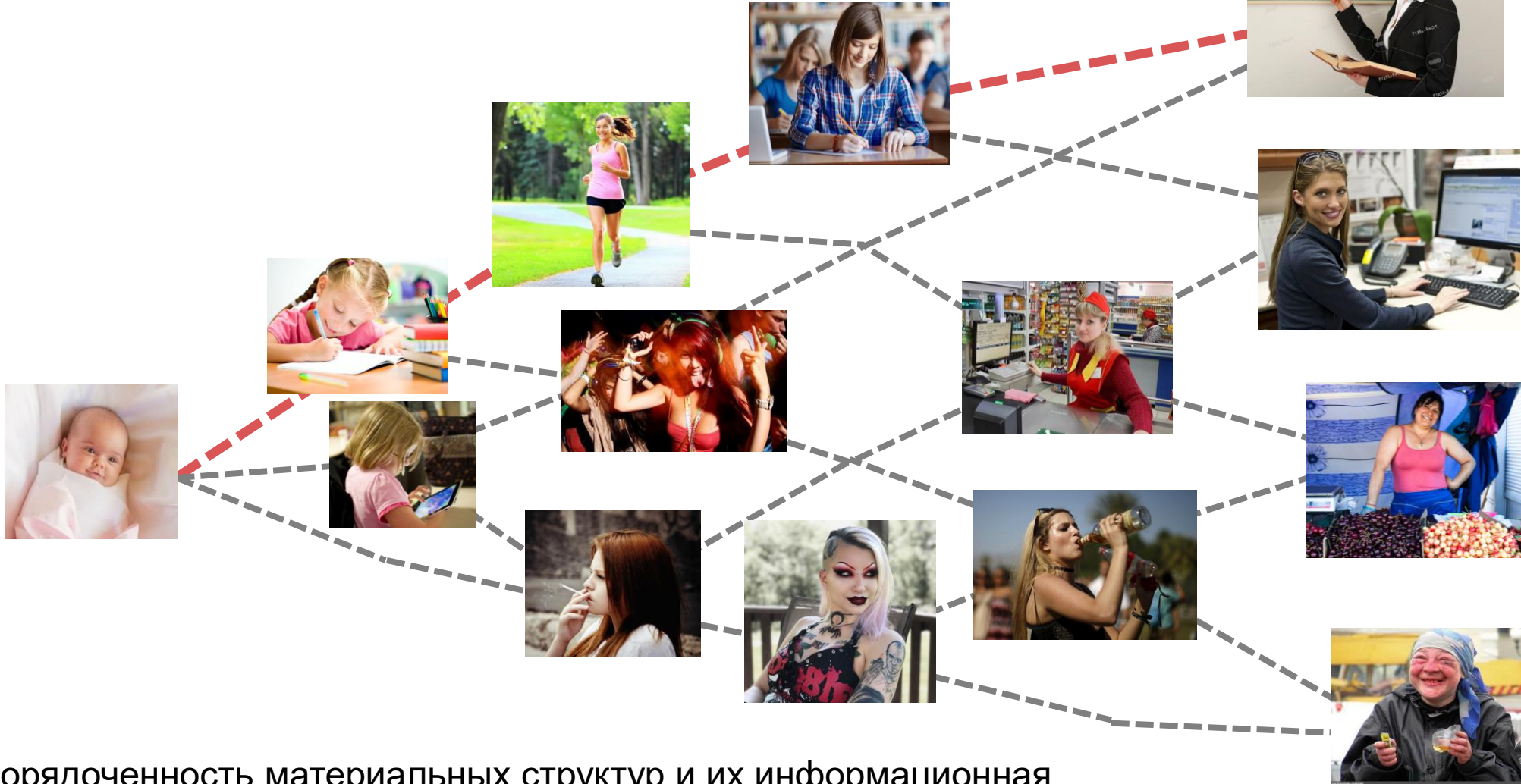
Общевселенская мера (матрица) — многовариантный сценарий бытия Мироздания, предопределяющий упорядоченность материальных структур (их информационную ёмкость) и пути их изменения.



Упорядоченность материальных структур и их информационная ёмкость изменяются при поглощении информации и при её потере.

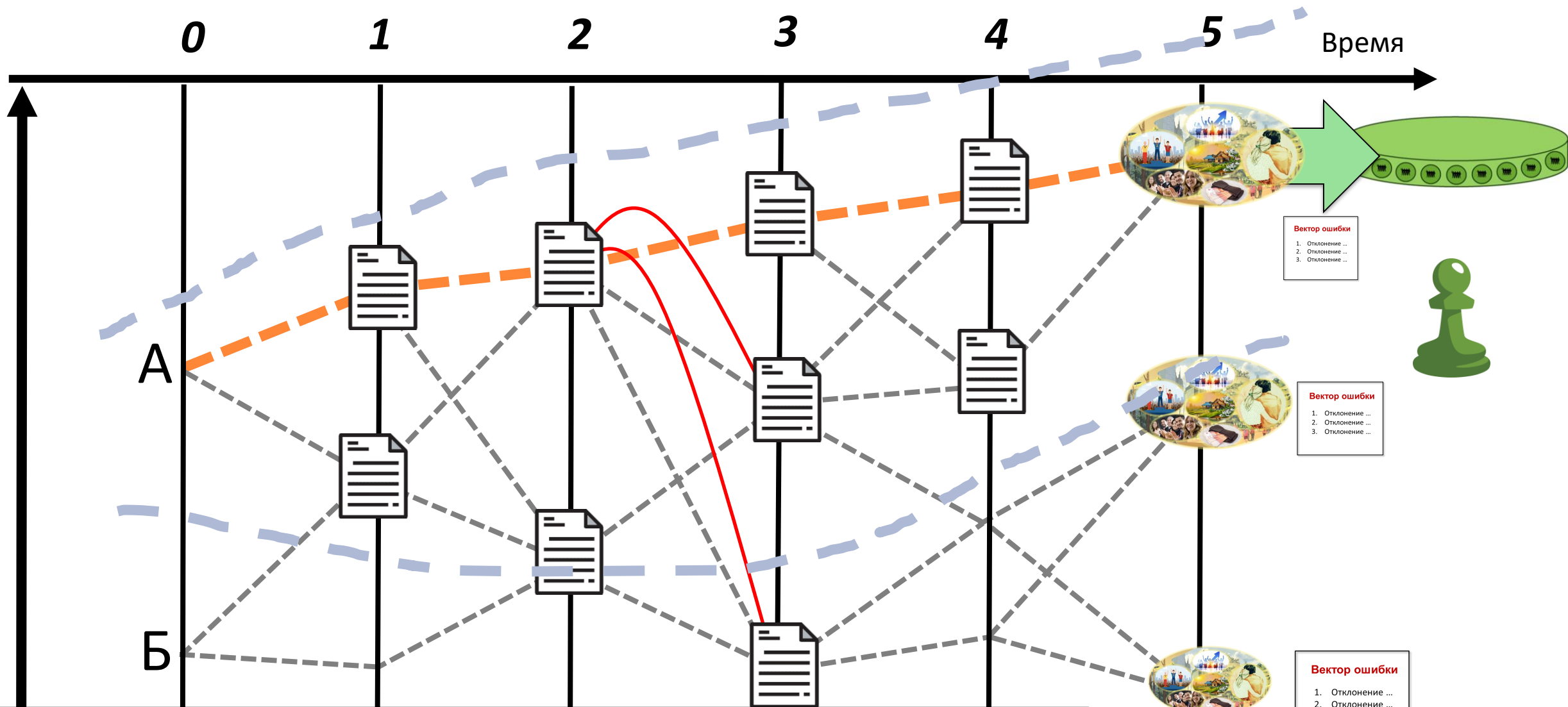
Матрица возможных состояний (судьба)

Общевселенская мера (матрица) — многовариантный сценарий бытия Мироздания, предопределяющий упорядоченность материальных структур (их информационную ёмкость) и пути их изменения.



Упорядоченность материальных структур и их информационная ёмкость изменяются при поглощении информации и при её потере.

Множество возможных состояний



Вектор ошибки
1. Отклонение ...
2. Отклонение ...
3. Отклонение ...

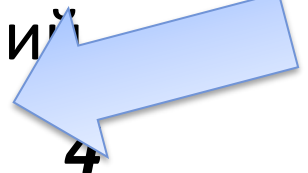
Вектор ошибки
1. Отклонение ...
2. Отклонение ...
3. Отклонение ...

Вектор ошибки
1. Отклонение ...
2. Отклонение ...
3. Отклонение ...

Вектор ошибки растёт

Если ваша МЕЧТА – это вхождение в объемлющий процесс поддерживаемый Свыше, то шаги к ней являются вашей ЗАЩИТОЙ и ОРУЖИЕМ против недоброжелателей. Продвижение по оптимальной траектории может восприниматься как редкостное везение, а отступление от неё как «возвращение на круги своя» или как полоса жестокого невезения.

Множество возможных состояний

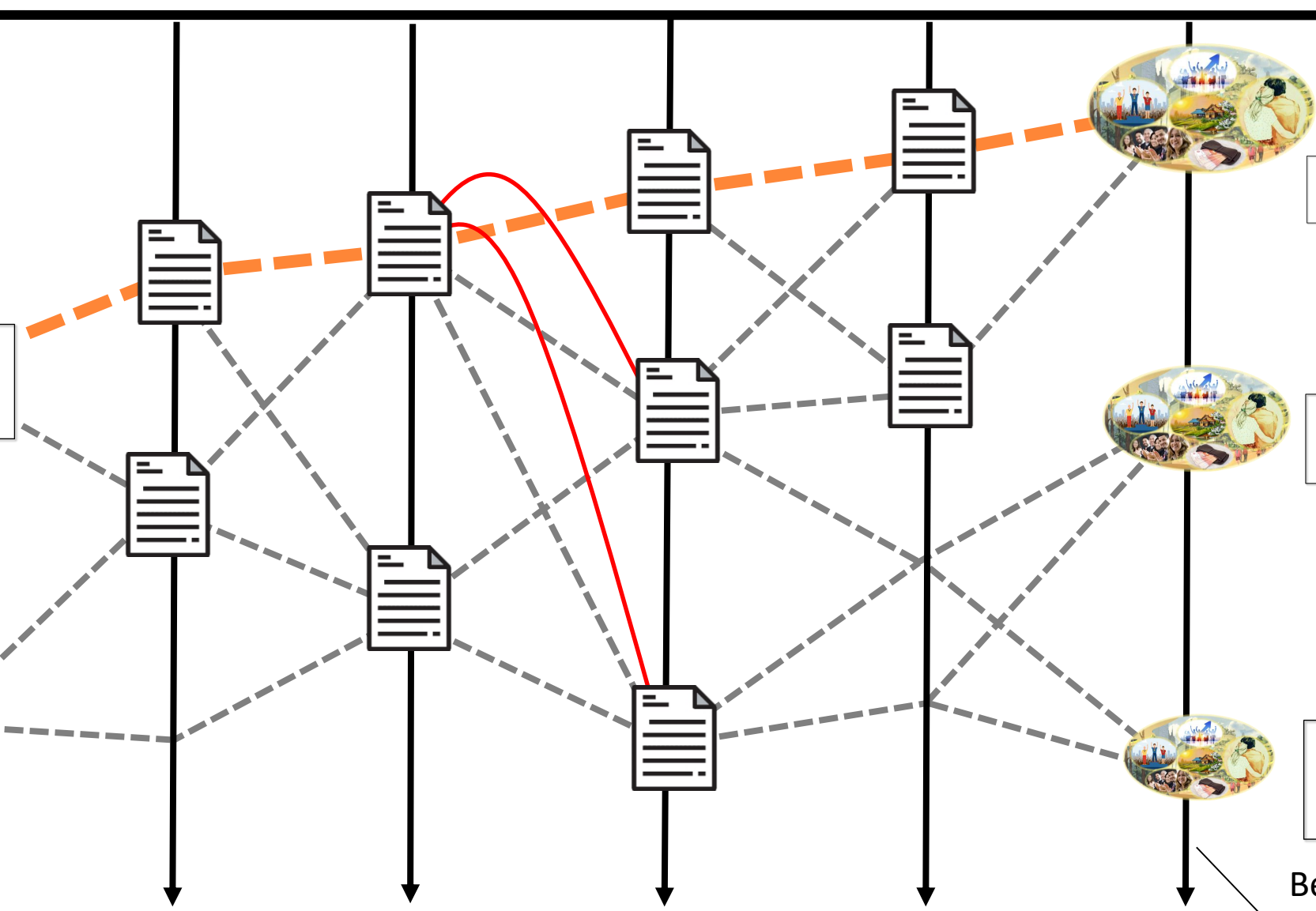


0 1 2 3 4 5 Время



Вектор текущего состояния
1. Имеет место ...
2. Имеет место ...
3. Имеет место ...

Б



Вектор ошибки
1. Отклонение ...
2. Отклонение ...
3. Отклонение ...

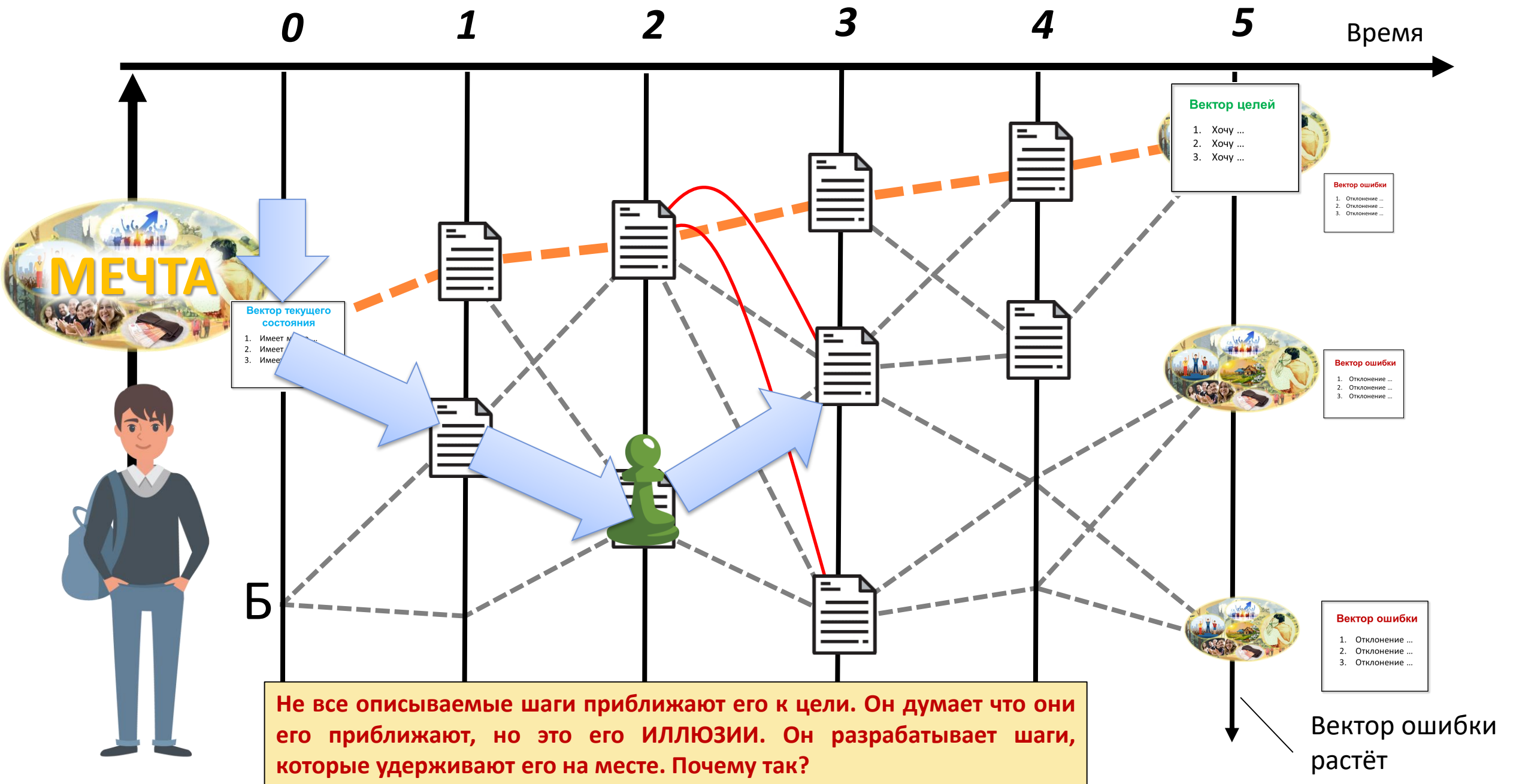
Вектор целей
1. Хочу ...
2. Хочу ...
3. Хочу ...

Вектор ошибки
1. Отклонение ...
2. Отклонение ...
3. Отклонение ...

Вектор ошибки
1. Отклонение ...
2. Отклонение ...
3. Отклонение ...

Вектор ошибки растёт

Множество возможных состояний



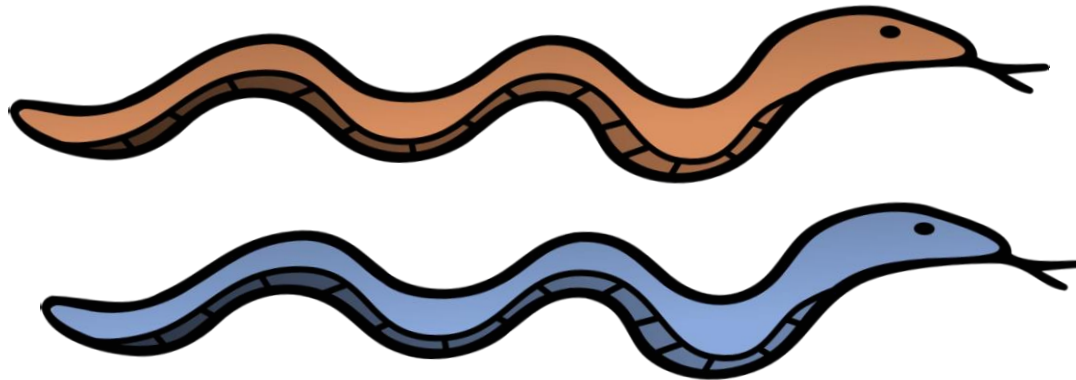


ШАГ содержащий выигрыш – это
соразМЕРный шаг



Выигрыша нет

➤ Если не **УЗНАЕШЬ**, то и не **СДЕЛАЕШЬ**



М
Е
Р
А

**ПРАКТИКА –
КРИТЕРИЙ ИСТИНЫ**

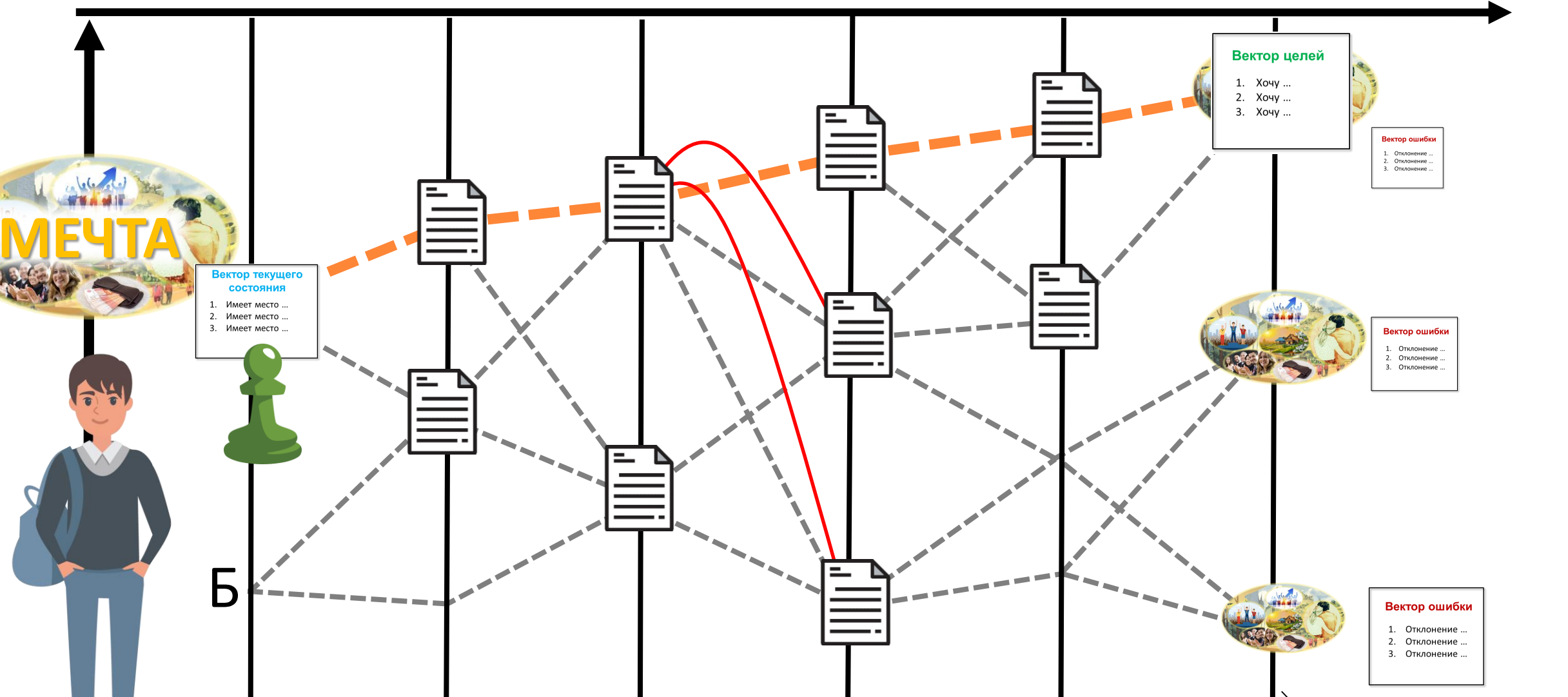
➤ Если не **ПОПРОБУЕШЬ**, то и не **УЗНАЕШЬ**

Множество возможных состояний



0 1 2 3 4 5

Время



Вектор текущего состояния
1. Имеет место ...
2. Имеет место ...
3. Имеет место ...

Вектор целей
1. Хочу ...
2. Хочу ...
3. Хочу ...

Вектор ошибки
1. Отклонение ...
2. Отклонение ...
3. Отклонение ...

Вектор ошибки
1. Отклонение ...
2. Отклонение ...
3. Отклонение ...

Вектор ошибки
1. Отклонение ...
2. Отклонение ...
3. Отклонение ...

Валера не представляет себе ПОБЕДНОЕ СОСТОЯНИЕ и не описывает состояния предшествующие этой ПОБЕДЕ – для него это скучно.

Вектор ошибки растёт



Б

Моя мечта:

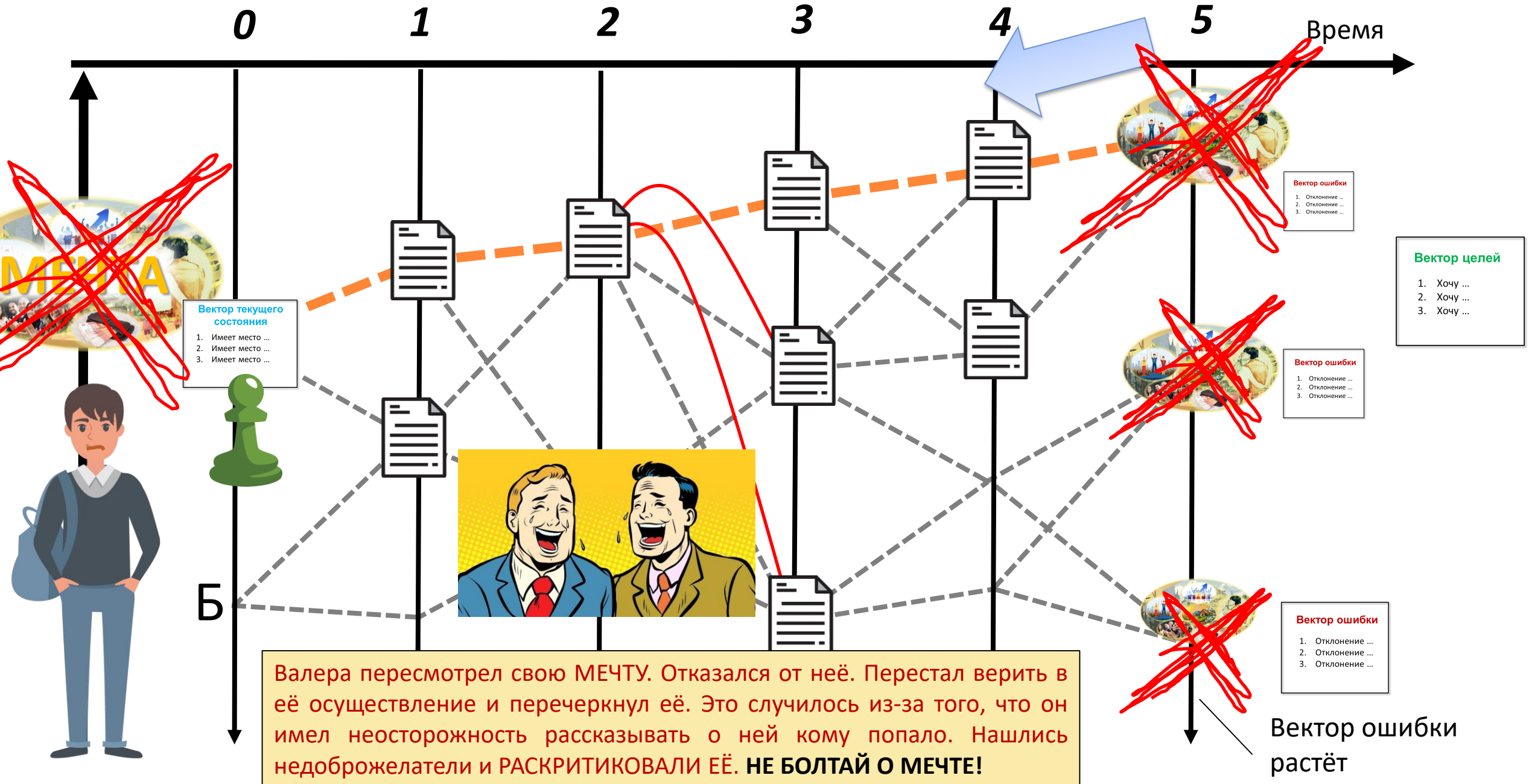
1. К 18 годам, я желаю достигнуть...
2. Я хочу стать...
3. Я хочу обрести...

К 16 годам,
у меня будет...
Я начну ...

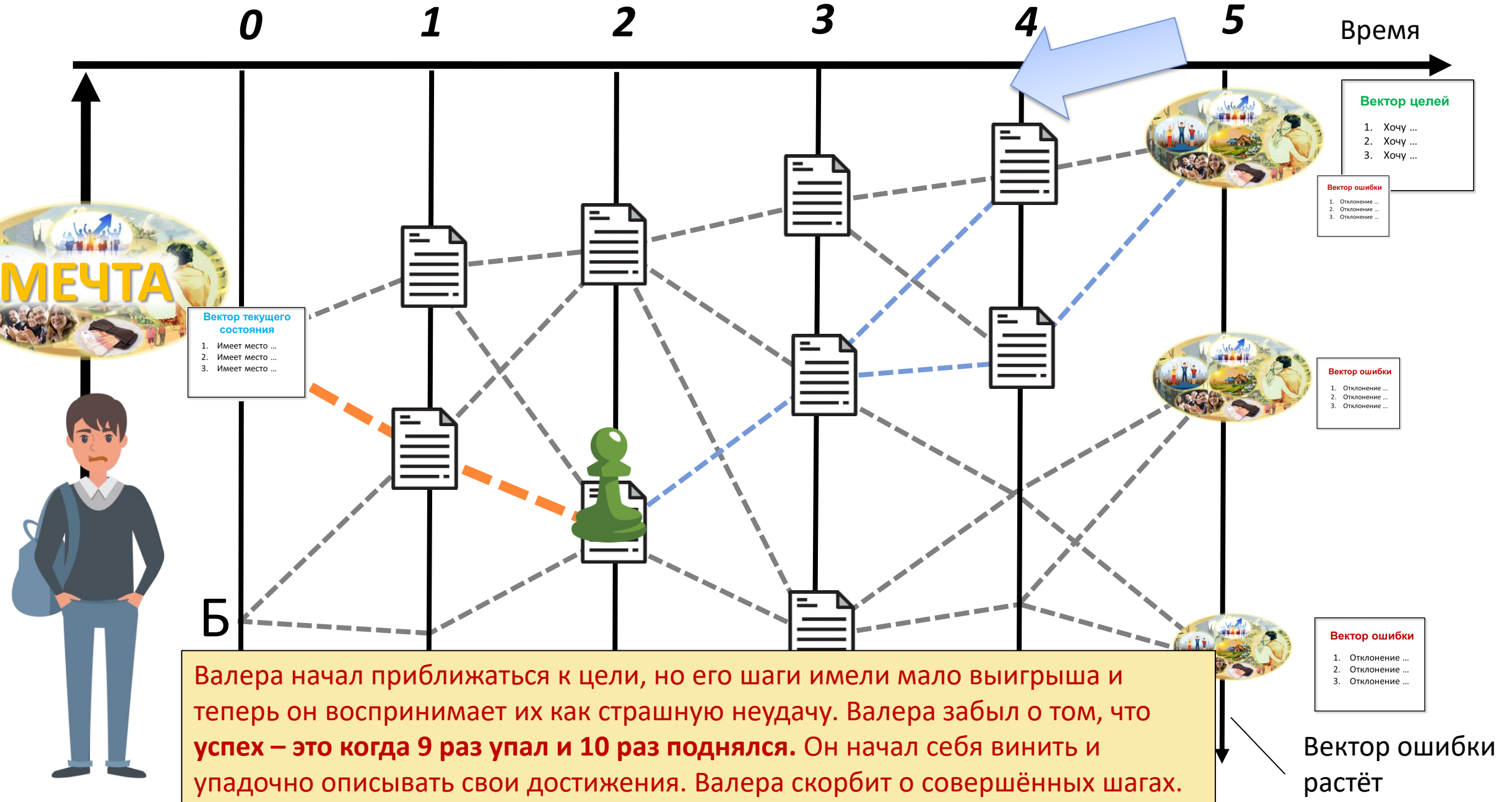
К 14 годам,
у меня будет...
Я начну ...



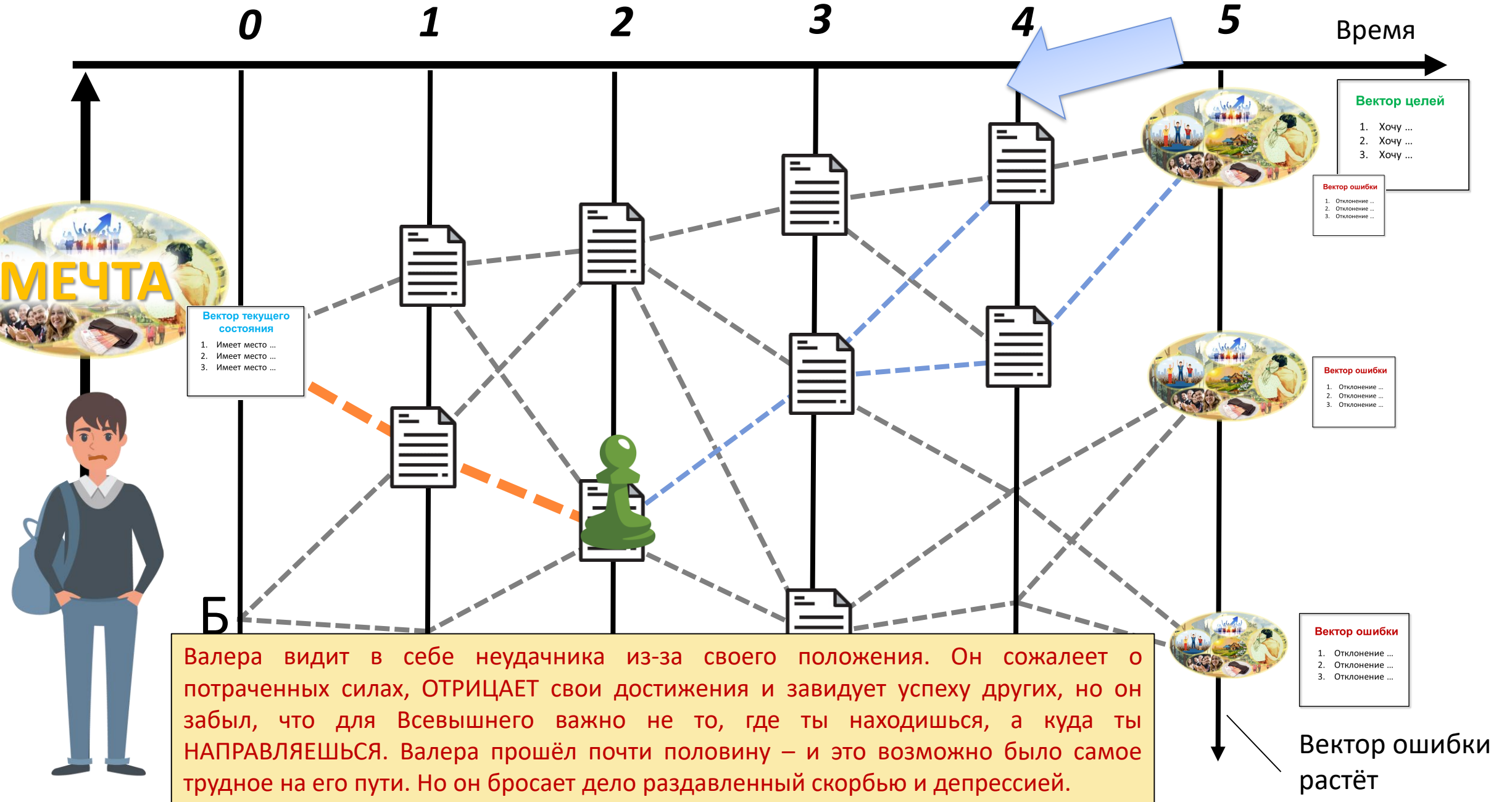
Множество возможных состояний



Множество возможных состояний



Множество возможных состояний



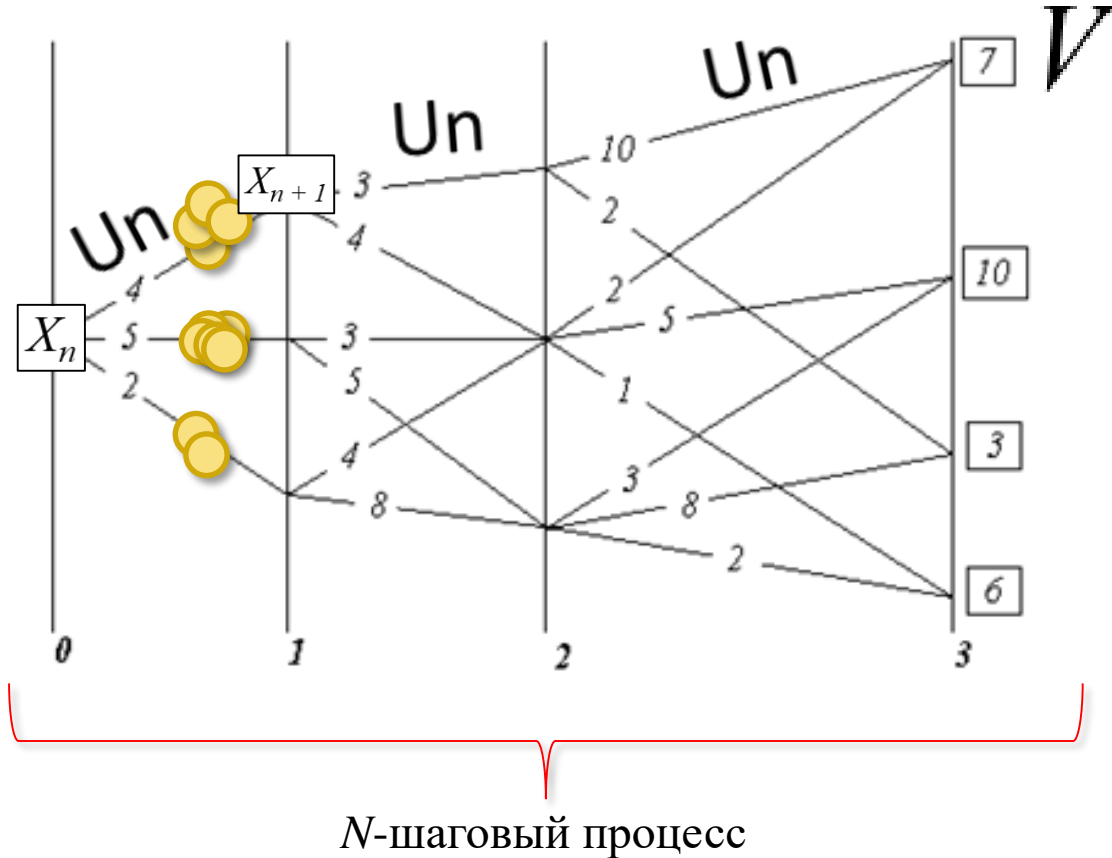
Правила метода динамического программирования

1. Достижение цели (желаемого состояния) рассматривается как **ПРОЦЕСС В НЕСКОЛЬКО ШАГОВ. ПРОЦЕСС УПРАВЛЕНИЯ ДЕЛИМ на ШАГИ**: каждый шаг – приближает к цели). Можно описать **МЕЧТУ** (будущее) и двигаться к **НАЧАЛУ** (настоящее), описывая состояния перед ней.
2. **СТРУКТУРА ЗАДАЧИ НЕ ДОЛЖНА ИЗМЕНЯТЬСЯ** при изменении расчётного количества шагов (**ЖЕЛАЕМАЯ МЕЧТА – НЕДОЛЖНА ПЕРЕСМАТРИВАТЬСЯ: НЕ ПРЕДАВАЙ МЕЧТУ**).
3. **ОПИСАНИЕ СОСТОЯНИЯ СИСТЕМЫ НЕДОЛЖНО ИЗМЕНЯТЬСЯ** от количества шагов (**ТВОЁ ОТНОШЕНИЕ К СЕБЕ и к ДЕЛУ НЕДОЛЖНО МЕНЯТЬСЯ, КАК БЫ НИ ДАЛЕКА и НИ ТРУДНА БЫЛА ЦЕЛЬ – «Я и ДЕЛО - НЕПОКОЛЕБИМЫ!»**). **НЕ ПРИНИЖАТЬ СЕБЯ и ДЕЛО!**
4. **НЕЛЬЗЯ ОТРИЦАТЬ ВЫБОР УПРАВЛЕНИЯ НА ПРЕДЫДУЩИХ ШАГАХ**. Следующий шаг должен определяться достигнутым состоянием, а не впечатлениями о «неудачах» сделанных шагов. *Не важно где ты находишься, важно то, к чему ты стремишься (ВАЖНЫ ТВОИ НАМЕРЕНИЯ)*. *Успех – это когда 9 раз упал, но 10 раз поднялся.*

Метод динамического программирования работоспособен, если формальная интерпретация реальной задачи позволяет выполнить следующие условия:

1. Рассматриваемая задача может быть представлена как N -шаговый процесс, описываемый соотношением:

$X_{n+1} = f(X_n, U_n, n)$, где n — номер одного из множества возможных состояний системы, в которое она переходит по завершении n -ного шага; X_n — вектор состояния системы, принадлежащий упомянутому n -ному множеству; U_n — управление, выработанное на шаге n (шаговое управление), переводящее систему из возможного её состояния в n -ном множестве в одно из состояний $(n+1)$ -го множества. Чтобы это представить наглядно, следует обратиться к рис. 4, о котором речь пойдет далее.



$$X_{n+1} = f(X_n, U_n, n),$$

f Управление (переход)
 Выигрыш на предыдущем этапе

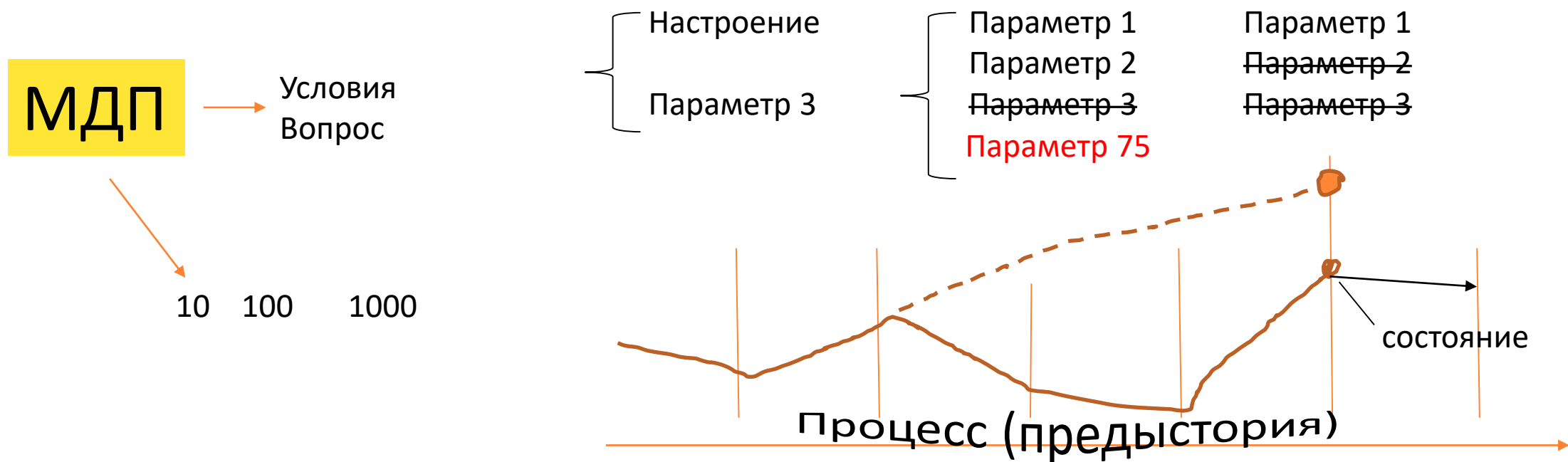
На рис. 4 показаны начальное состояние системы — «0» и множества её возможных последующих состояний — «1», «2», «3», а также возможные переходы из каждого возможного состояния в другие возможные состояния. Всё это вместе похоже на карту настольной детской игры, по которой перемещаются фишки: каждому переходу-шагу соответствует свой шаговый выигрыш, а в завершающем процессе третьем множестве — каждому из состояний системы придана его оценка, помещенная в

2. Структура задачи не должна изменяться при изменении расчетного количества шагов N .

3. Размерность пространства параметров, которыми описывается состояние системы, не должна изменяться в зависимости от количества шагов N .

4. Выбор управления на любом из шагов не должен отрицать выбора управления на предыдущих шагах. Иными словами, оптимальный выбор управления в любом из возможных состояний должен определяться параметрами рассматриваемого состояния, а не параметрами процесса, в ходе которого система пришла в рассматриваемое состояние.

Чисто формально, если одному состоянию соответствуют разные предыстории его возникновения, влияющие на последующий выбор оптимального управления, то метод позволяет включить описания предысторий в вектор состояния, что ведёт к увеличению размерности вектора состояния системы. После этой операции то, что до неё описывалось как одно состояние, становится множеством состояний, отличающихся одно от других компонентами вектора состояния, описывающими предысторию процесса.



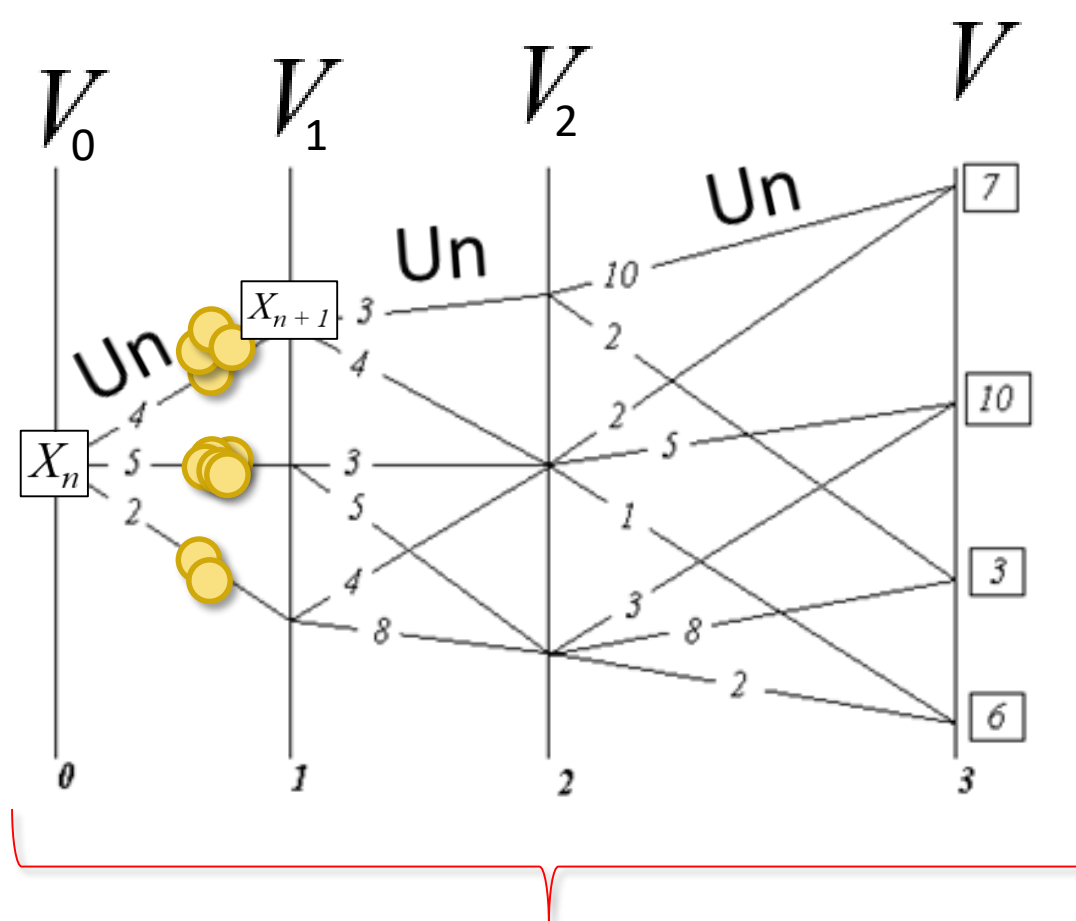
5. Критерий оптимального выбора последовательности шаговых управлений U_n и соответствующей траектории в пространстве формальных параметров имеет вид:

$$V = V_0(X_0, U_0) + V_1(X_1, U_1) + \dots + V_{N-1}(X_{N-1}, U_{N-1}) + V_N(X_N).$$

Критерий V принято называть *полным выигрышем*, а входящие в него слагаемые — *шаговыми выигрышами*. В задаче требуется найти *последовательность шаговых управлений* U_n и траекторию, которым соответствует максимальный из возможных *полных выигрышей*. По своему существу полный “выигрыш” V — мера качества управления *процессом в целом*. Шаговые выигрыши, хотя и входят в меру качества управления процессом в целом, но *в общем случае* не являются мерами качества управления на соответствующих им шагах, поскольку метод предназначен для оптимизации управления процессом в целом, а *эффективные шаговые управления* с большим шаговым выигрышем, но лежащие вне *оптимальной траектории*, интереса не представляют. Структура метода не запрещает при необходимости на каждом шаге употреблять критерий определения шагового выигрыша V_n , отличный от критериев, принятых на других шагах.

С индексом n — указателем-определителем множеств возможных векторов состояния — в реальных задачах может быть связан некий изменяющийся параметр, например: *время, пройденный путь, уровень мощности, мера расходования некоего ресурса* и т.п. То есть метод применим не только для оптимизации управления процессами, длящимися во времени, но и к задачам оптимизации многовариантного одномоментного или нечувствительного ко времени решения, если такого рода “безвременные”, “непроцессные” задачи допускают их многошаговую интерпретацию.

$$V = V_0(X_0, U_0) + V_1(X_1, U_1) + \dots + V_{N-1}(X_{N-1}, U_{N-1}) + V_N(X_N).$$



N -шаговый процесс

$$V = V_0(X_0, U_0) + V_1(X_1, U_1) + \dots + V_{N-1}(X_{N-1}, U_{N-1}) + V_N(X_N).$$