

# 「Linuxとイノベーション」

Jan. 30, 2009

Takao Ikoma  
シャープ株式会社  
研究開発本部プラットフォーム開発センター  
ikoma@slab.tnr.sharp.co.jp

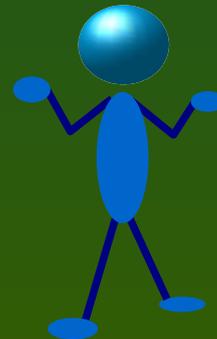
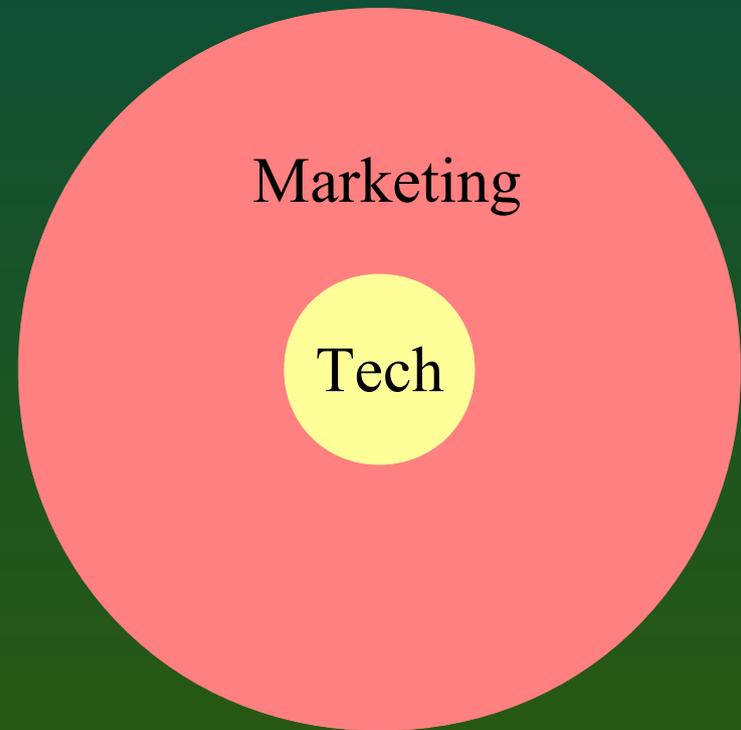
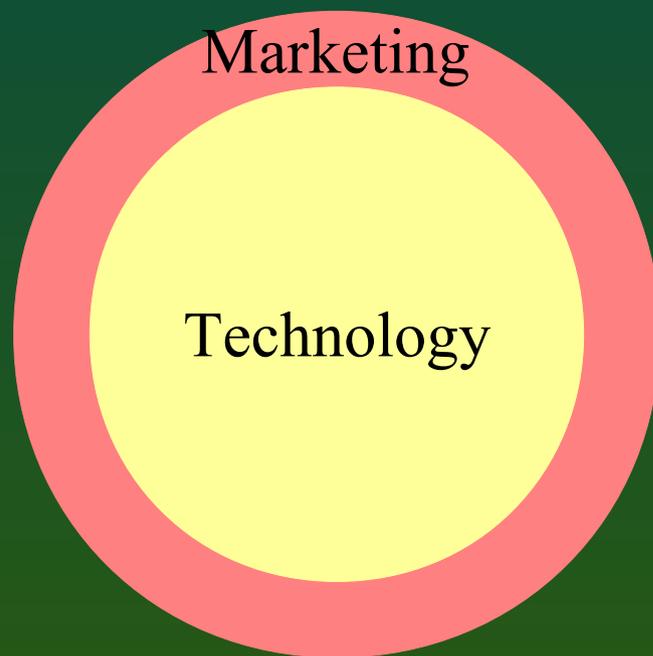
# 自己紹介

- 一応、ずっとソフト屋のつもり
  - オフコン、スーパーミニコン、PDAなど
  - 「おまえは製品のコードは書くな」と上司にいわれた
- やってきたこと
  - 開発環境やツールの開発、整備
  - ソフトウェアの品質向上
  - 技術企画
    - 社外技術のウォッチと導入、ベンチャービジネスとの付き合い、標準化、業界団体
- 自慢:オリジナルMacを自分の金で買った(1985)

# 技術以外の問題

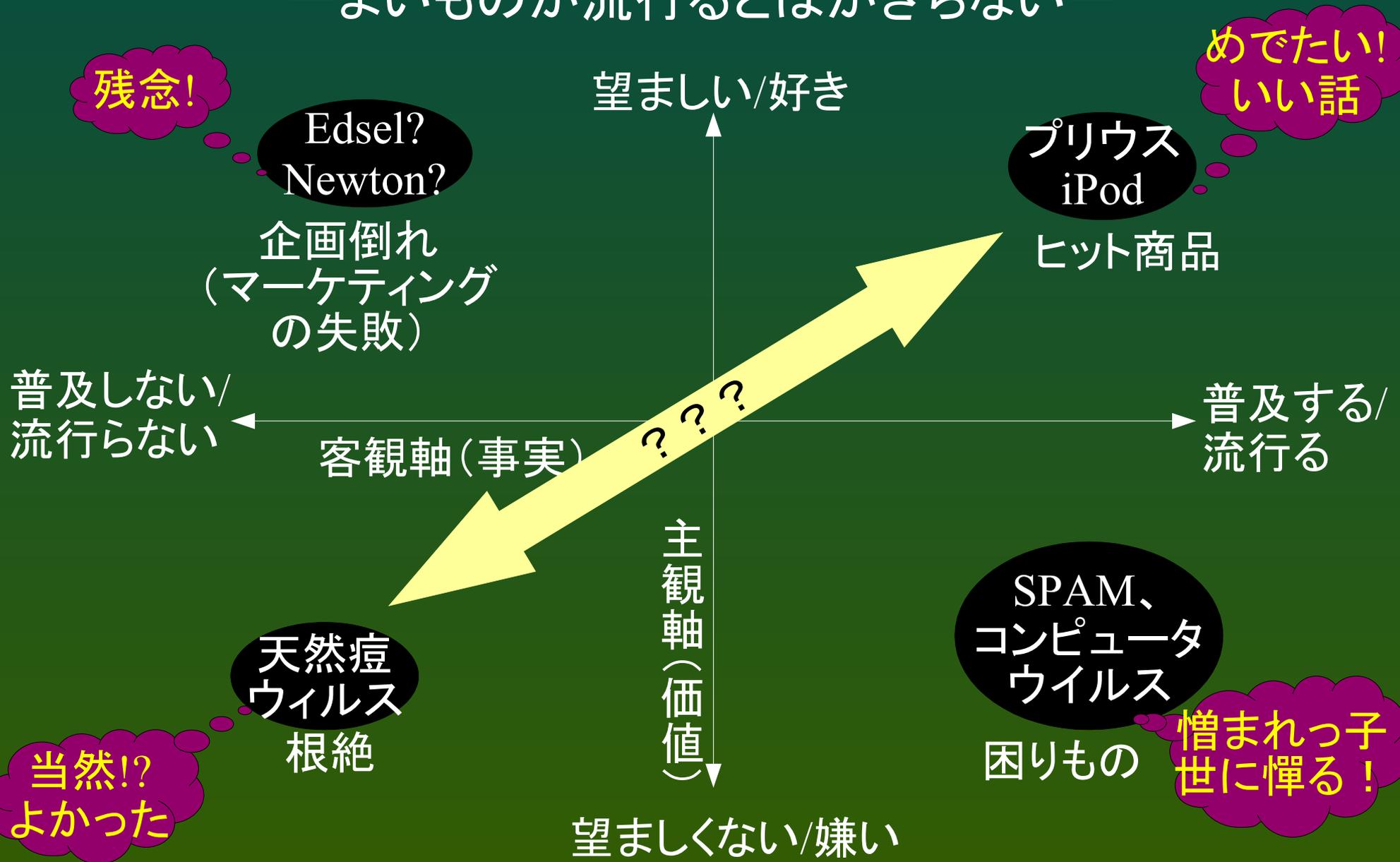
- 「おもしろい」だけでは「ビジネス」にならない
- 技術がいくらすぐれていても、それだけでは世の中に広まらない
  - 世の中に広めるための仕掛けが必要
    - ユーザニーズを満たす「完全なる商品」
  - それ以外の技術が主流になる可能性がある
  - その普及を妨害しようとするとする勢力があるかもしれない
- 今、ベストの技術を採用していても、次の世代で足元をすくわれる危険がある

# 外からはわからない！

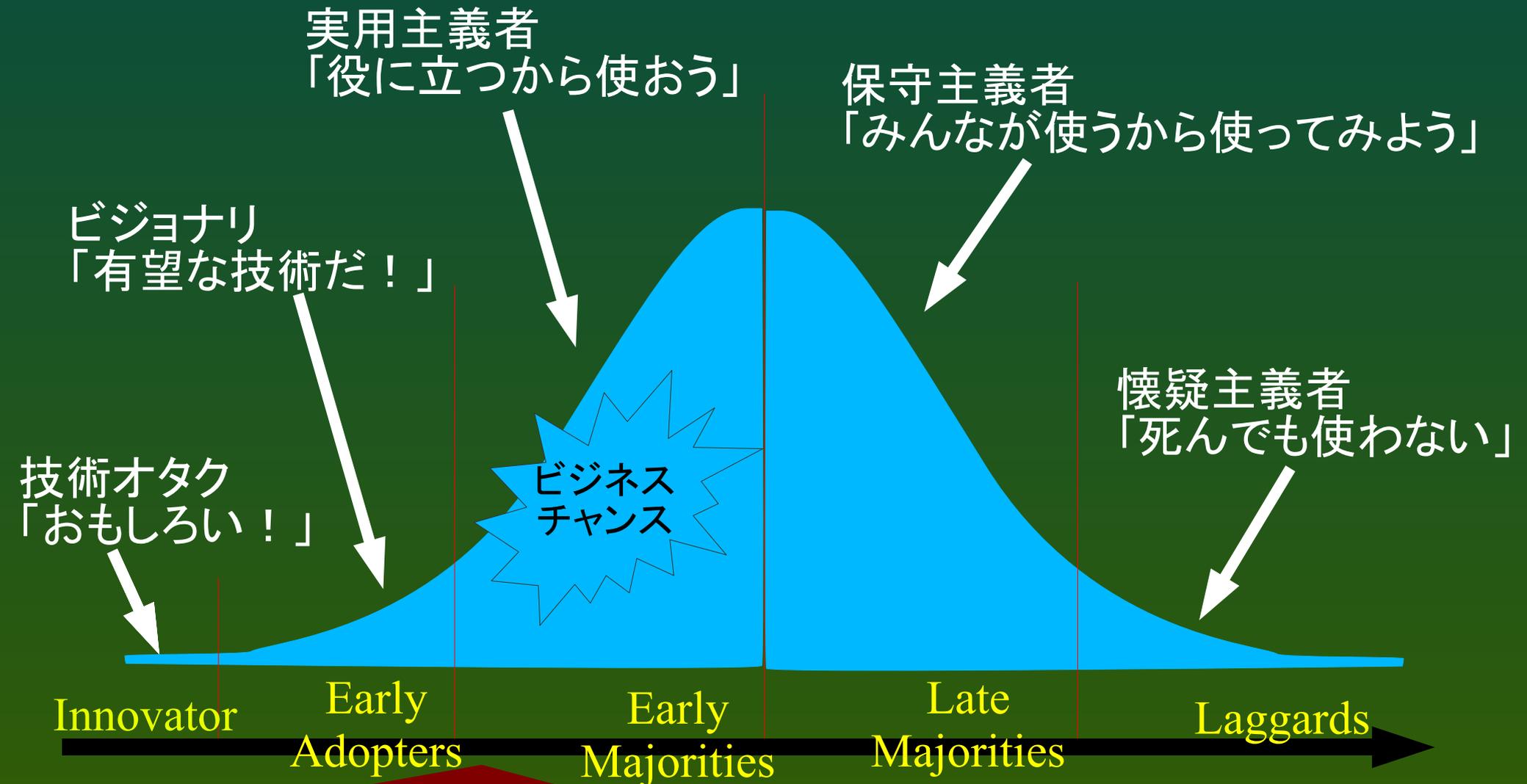


# 普及する？ しない？はミームが決める

—よいものが流行るとはかぎらない—



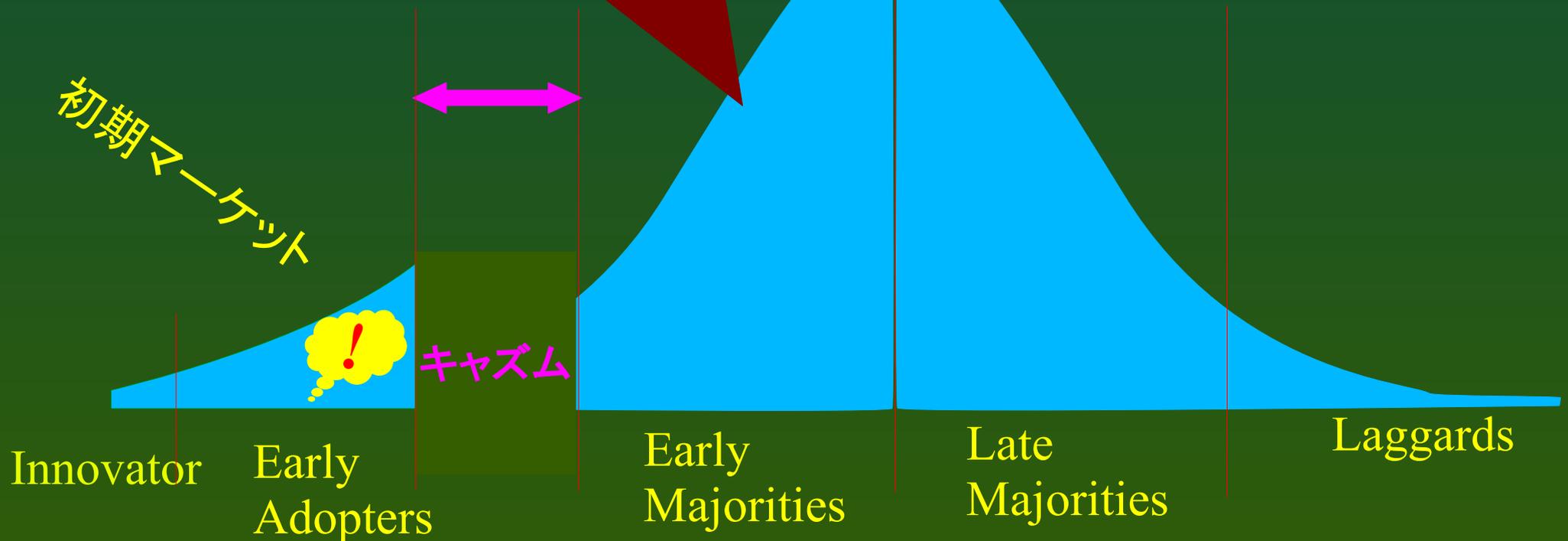
# テクノロジー採用ライフサイクル (Geoffrey A. Moore)



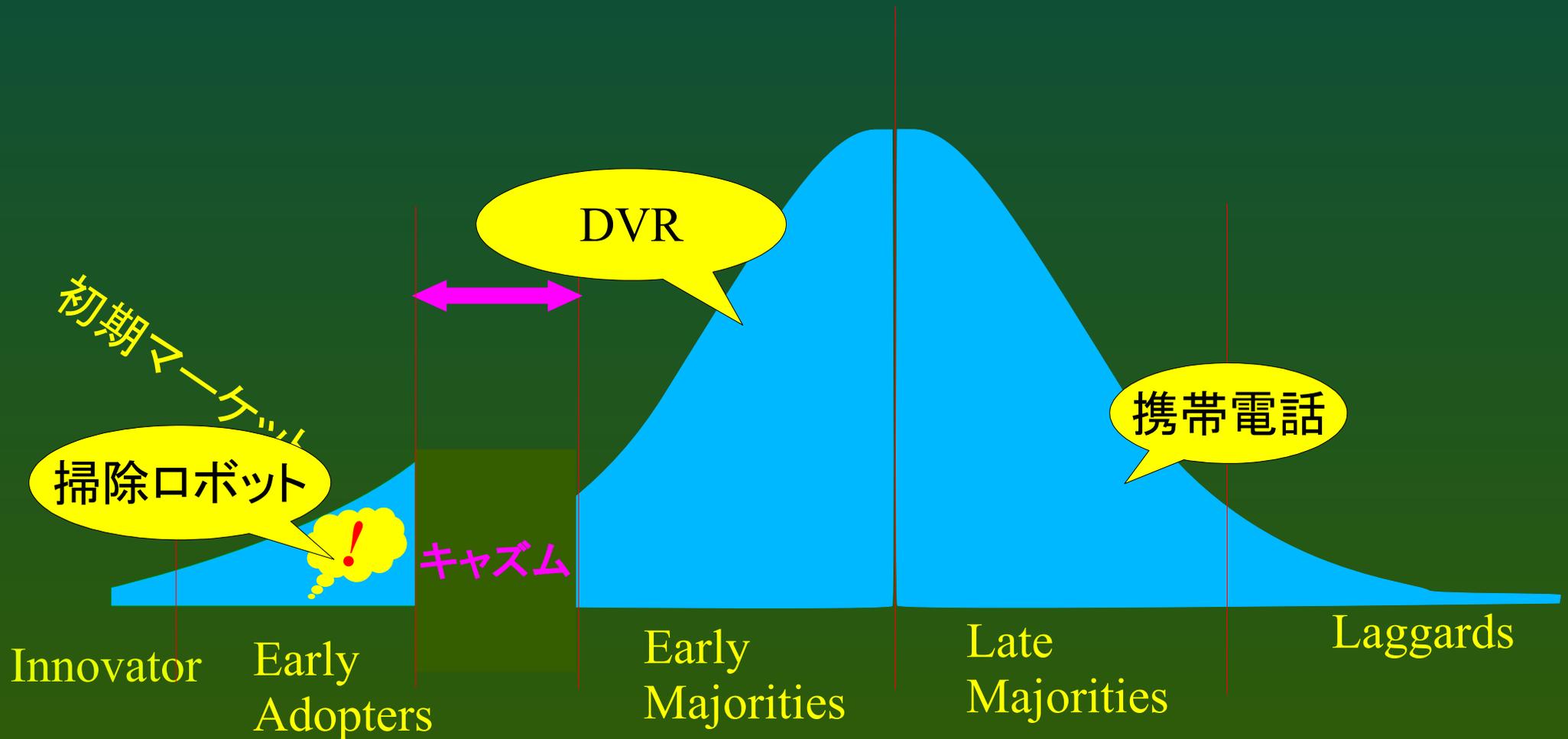
みんながいっせいに使い始めるのではない

# キャズム(死の谷)

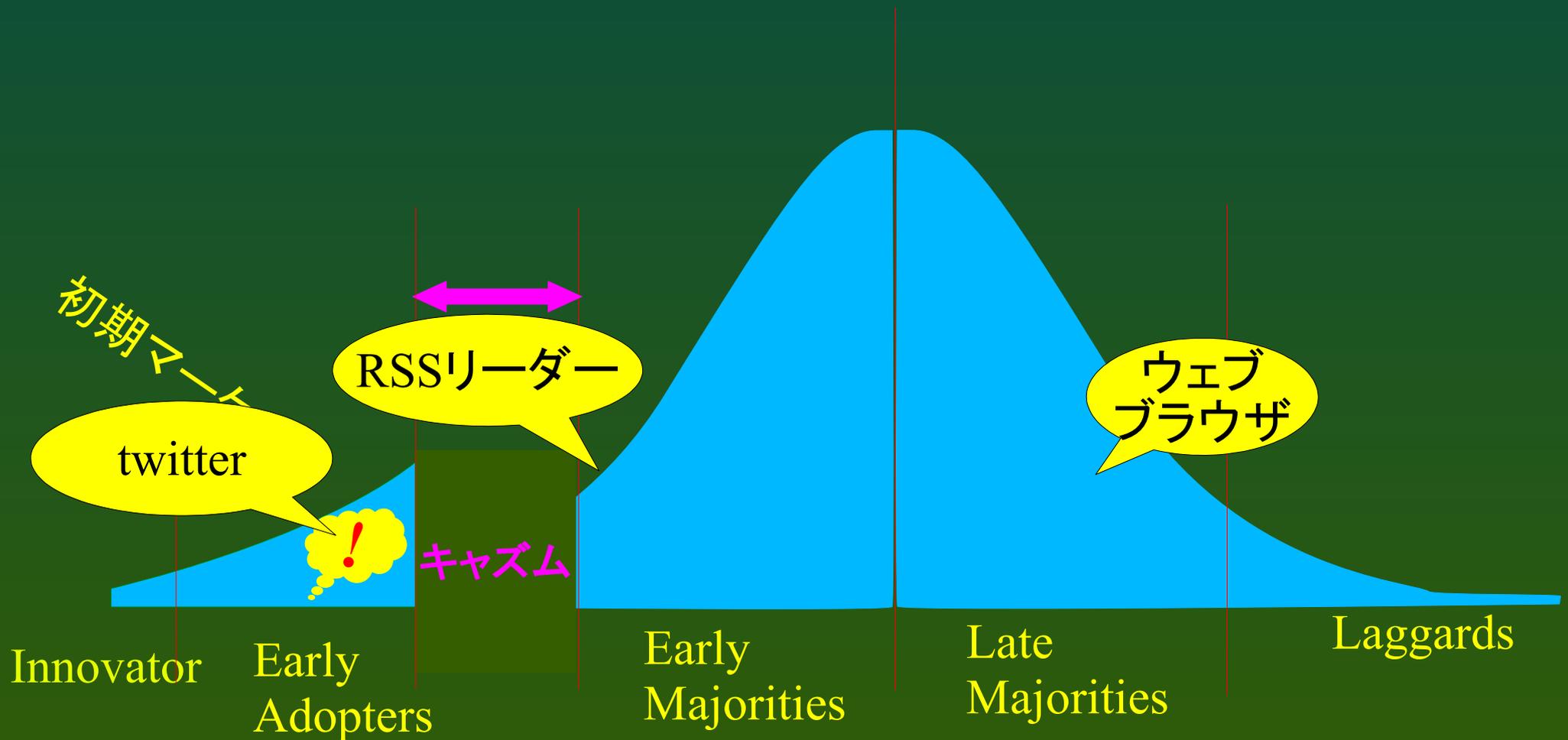
少数派が使い始めても  
多数派はついてこない  
("Show me cash!")



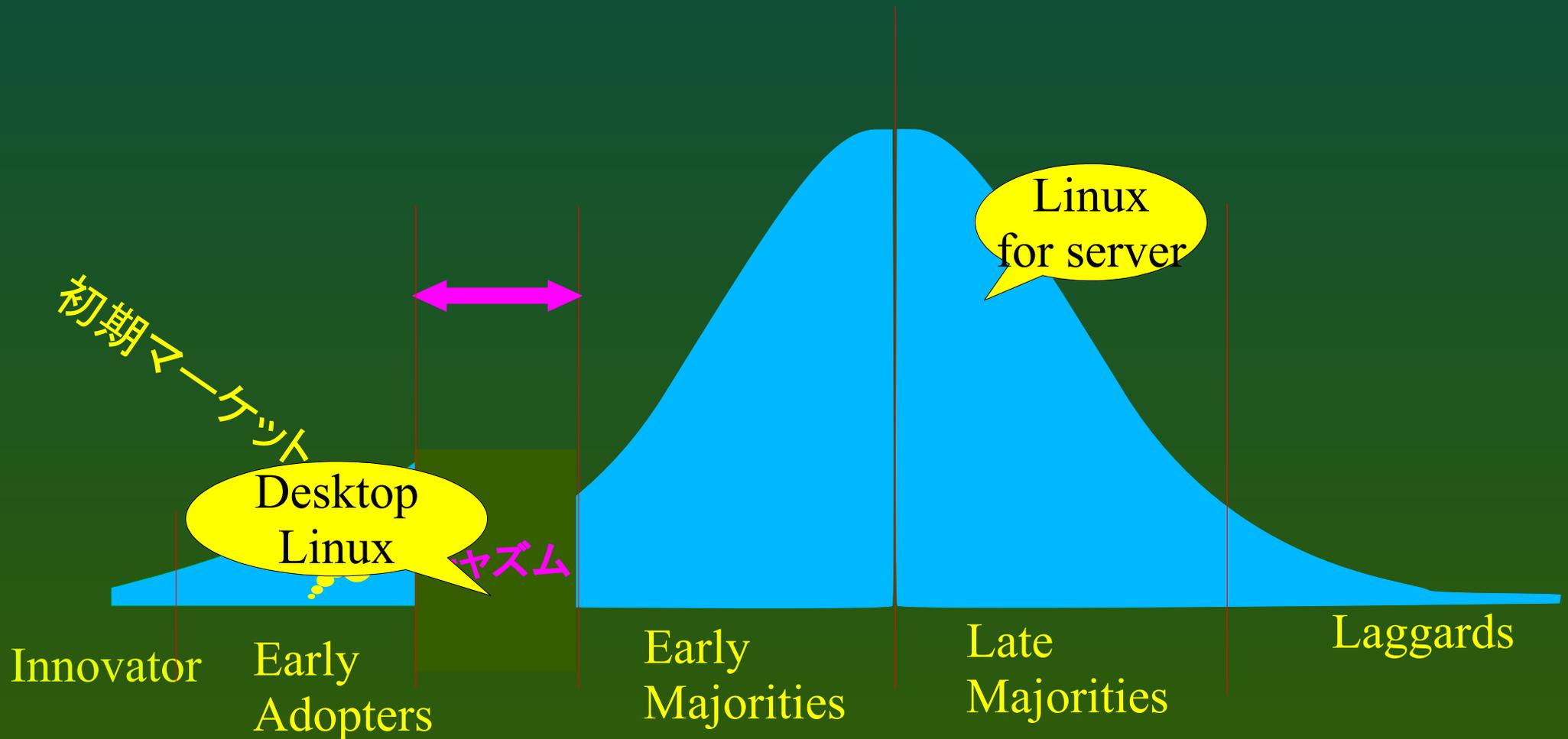
# キャズム(死の谷)



# キャズム(死の谷)

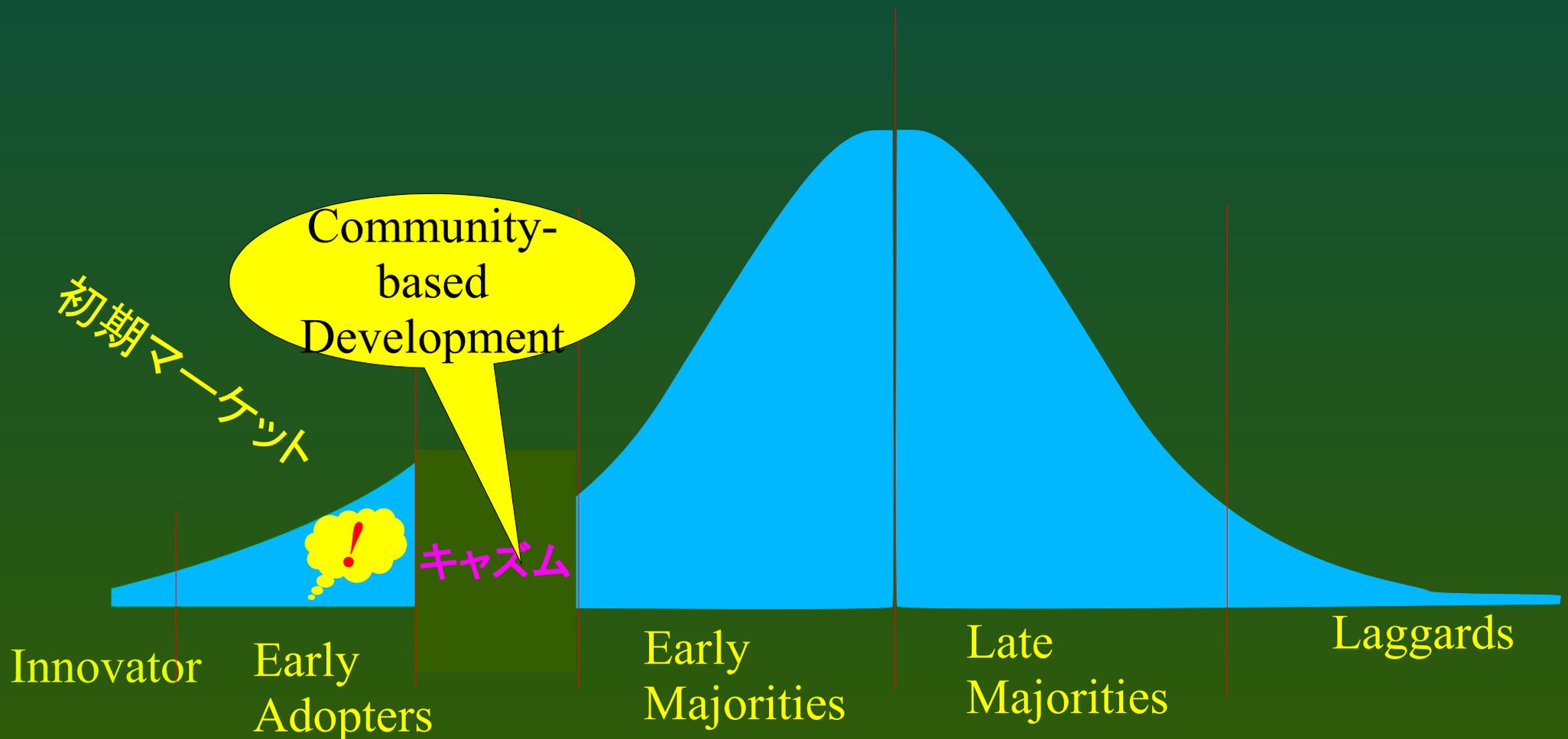


# キャズム(死の谷)



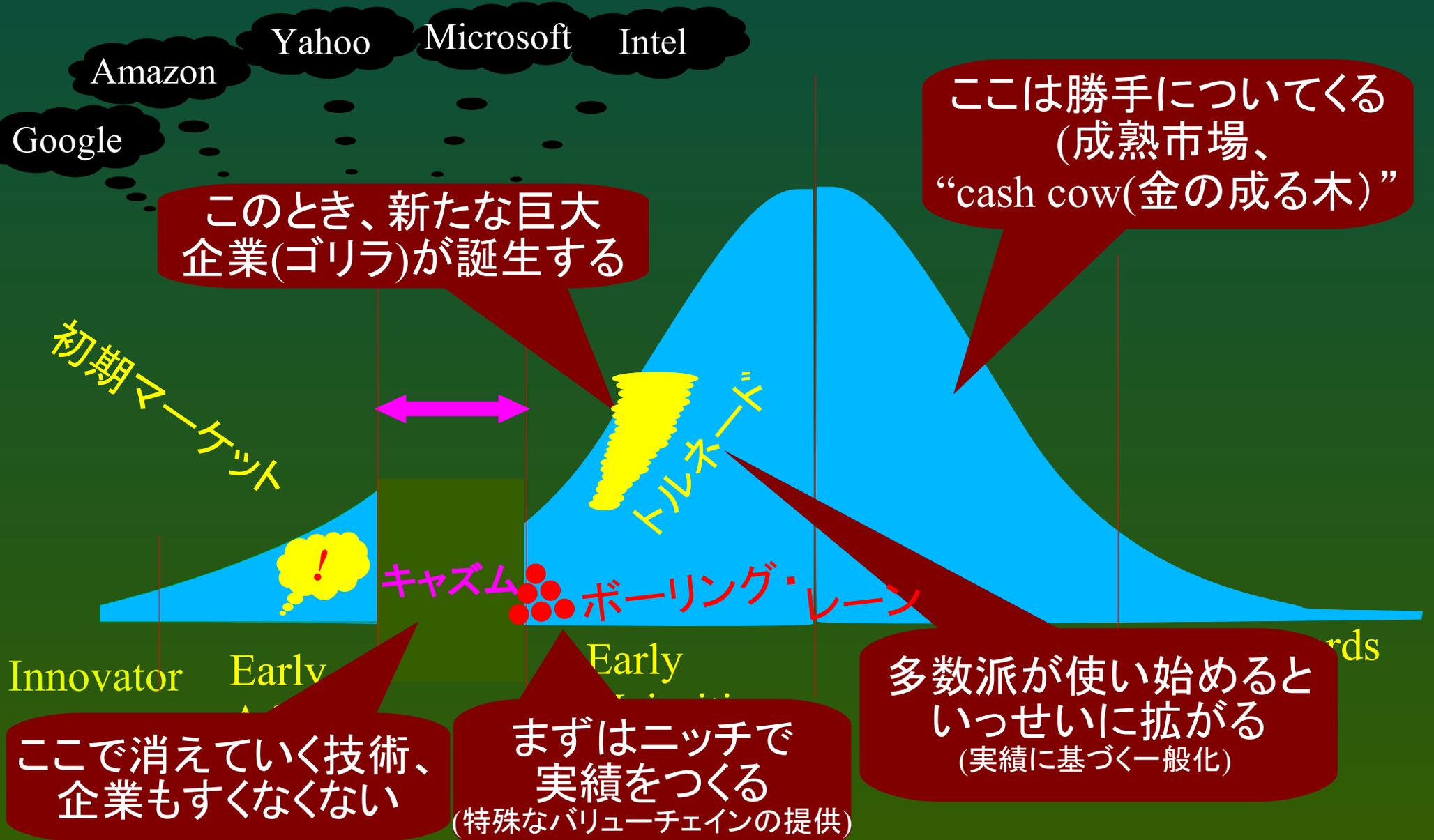


# キャズム(死の谷)



キャズムを越えるには: バリュー・チェーン(価値連鎖)が必要

# トルネード: 「ゴリラ」の誕生



# キャズムを越える

困って(いて、お金をもっている)のは誰？(課題の発見)

- 解決策(=ソリューション、「完全なる商品」、バリューチェーン)を提案する
- 高くて/大きすぎて/難しすぎて、買えない/使えないものでも事情があれば買って/使ってもらえるかもしれない
- ニッチでの実績がマスマーケットにつながる(「ボーリング・レーン」)
- 最初にキャズムを越えることができれば、急成長できる可能性が高い(「トルネード」、「ゴリラ」の誕生)

# 完全なる商品("the Whole Product")

- ユーザが望むものは裸のハードウェアだけではない
  - ソフトウェア(OS、アプリケーション、...)
  - 周辺装置、アクセサリ(ストレージ、ネットワーク、ケーブル、カメラ、...)
  - ネットワーク・インフラ、サービス、コンテンツ
  - 開発環境、トレーニング、サポート
  - システム・インテグレーション
- 「Linux」だけではソリューションにはならない
- **バリューチェーン(価値連鎖)の提供**
  - 「困っている管理者」の問題を解決してあげる(ソリューション)
  - もはや1社ですべて提供することは不可能→協業

# iPodのバリュー・チェーン



Value Chain

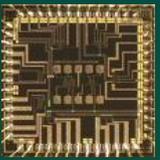
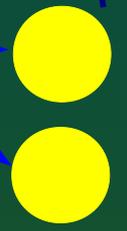
Value Proposition(ユーザの求めているもの)

artists,  
composer,

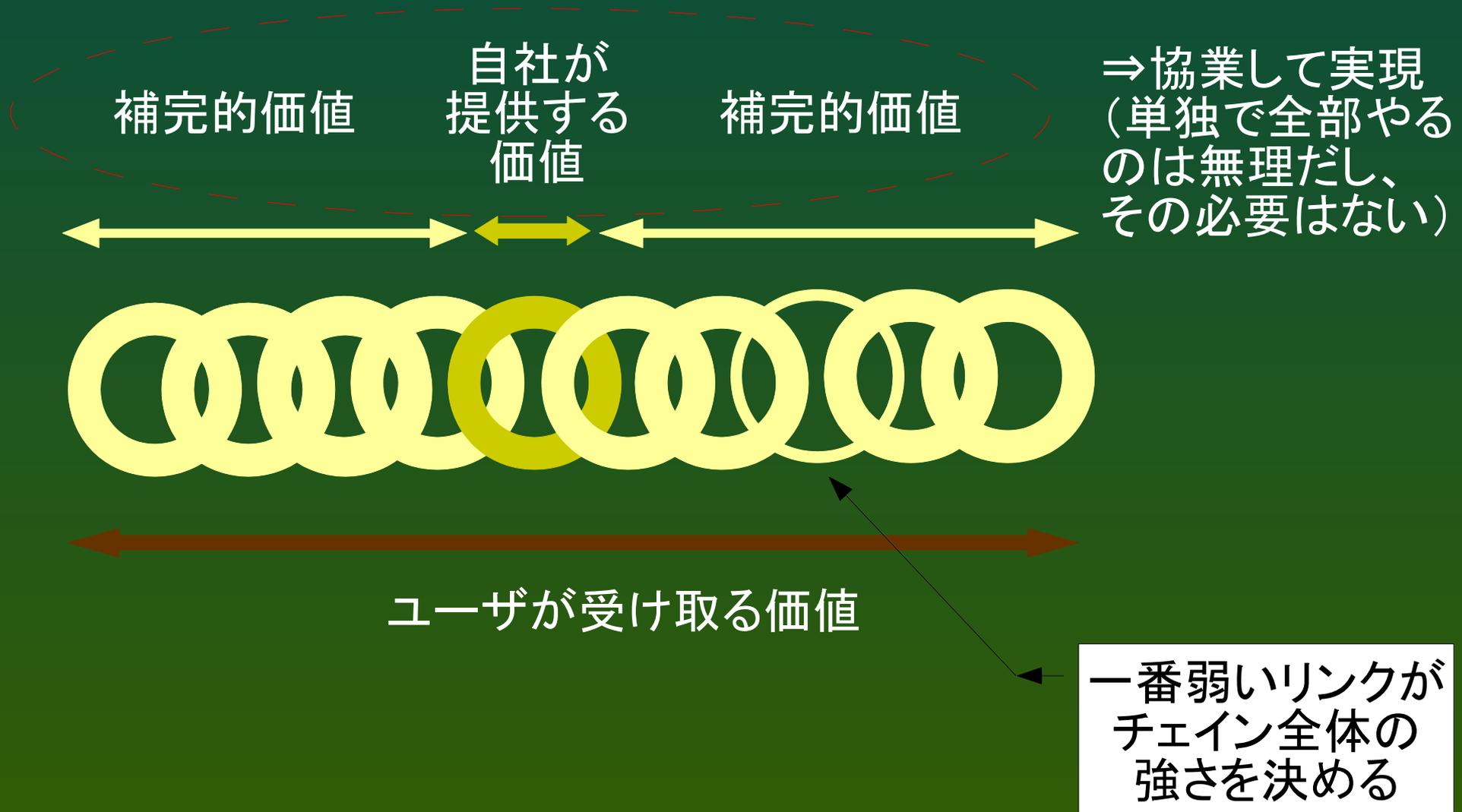
Contents providers  
(Universal,  
Disney,  
etc.)

TV Drama

Podcasting



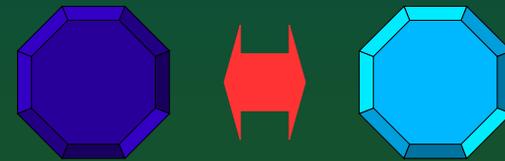
# バリューチェーン(価値連鎖) あるいはエコシステム(生態系)



# 競争の変遷

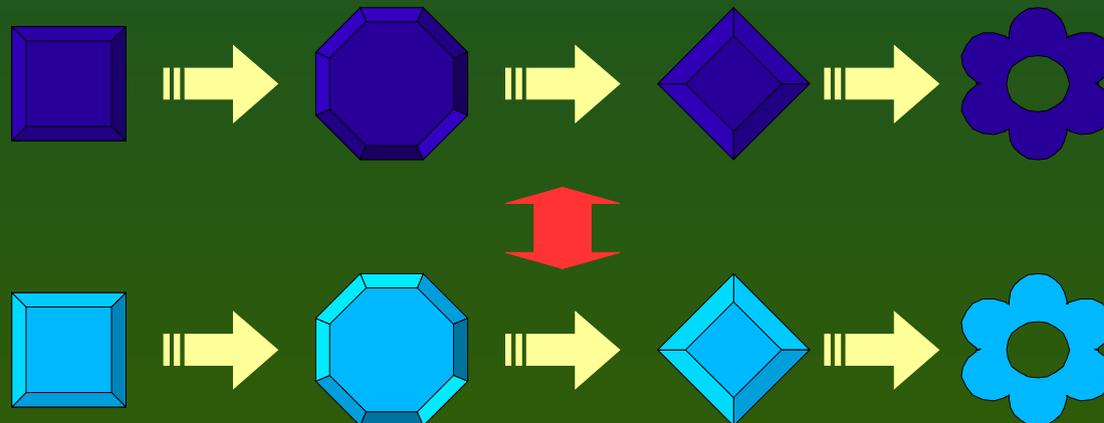
- 企業 vs 企業 の競争

- e.g. Panasonic vs Sharp



- バリューチェーン vs バリューチェーンの競

- e.g. iPod vs CD、Windows vs Linux



# フリーライダーと社会的ジレンマ

- 囚人のジレンマ
  - 二人、コミュニケーションなし

	協力	非協力
協力	2, 2	5, 0
非協力	0, 5	4, 4

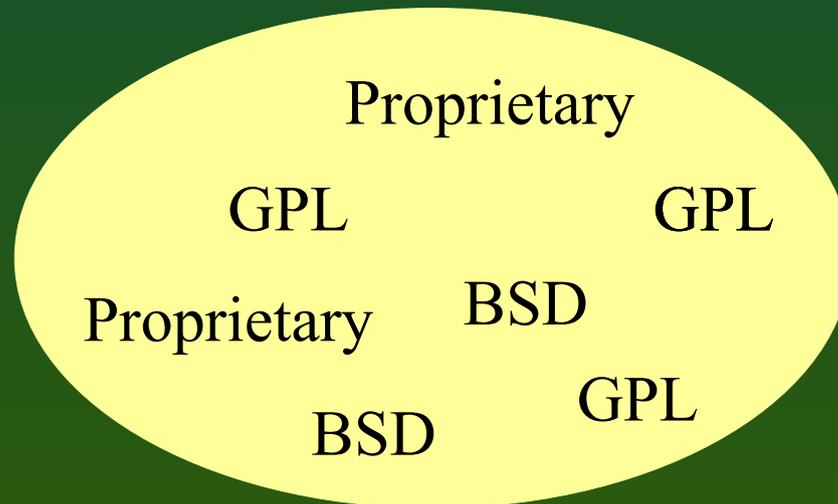
フリーライダー



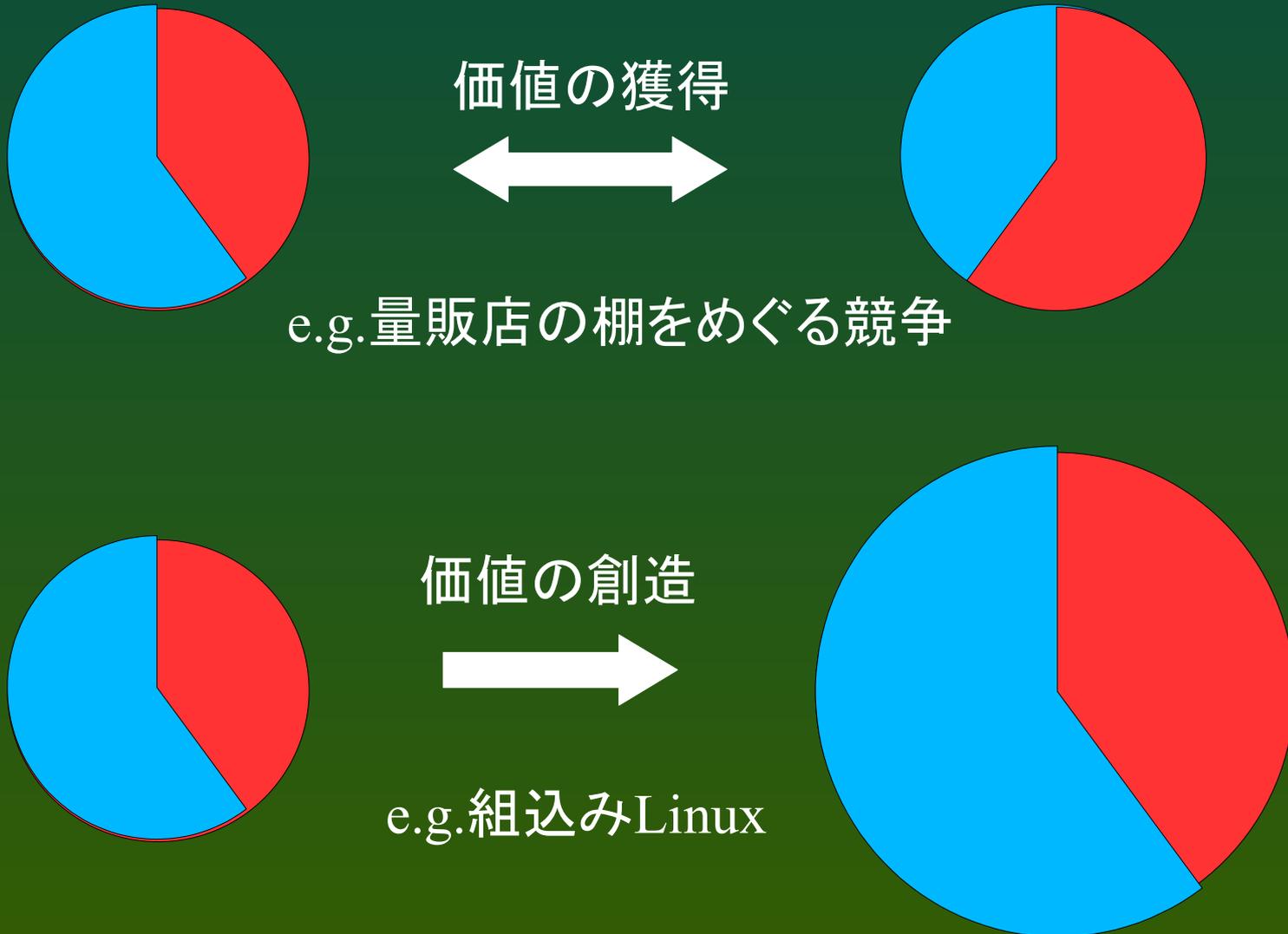
- 社会的ジレンマ、共有地の悲劇
  - 多人数ゲーム

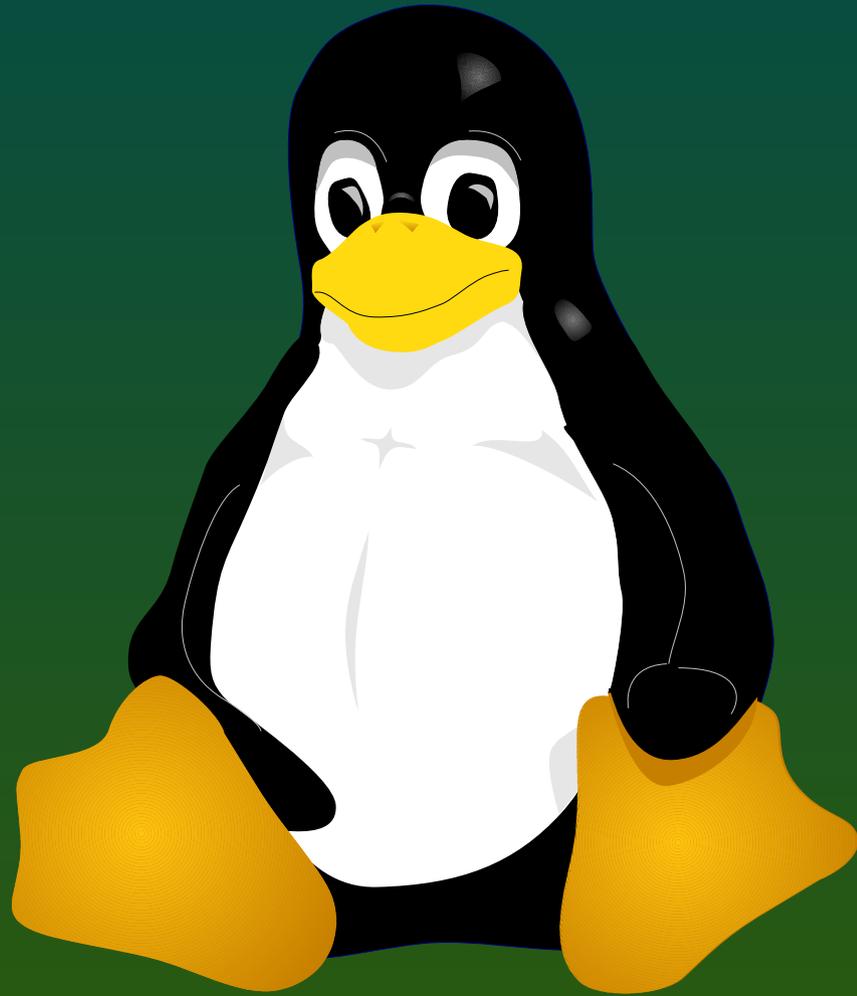
# GPL

- フリーライダーを抑制するしくみ
- 進化的に安定な戦略(ESS)???



# Zero SumゲームとWinWinゲーム





# イノベーターのジレンマ

(Clayton Christensen)

- 現在の技術の延長上ではないところから新しい技術が登場し、主役を奪ってしまう
- 現状で最適なオペレーションが命取り
  - 利益率の薄いローエンドでの競争を逃げていると、ニッチプレイヤーに追いやられる
- 今日の「勝ち組」が明日の「負け組」になる

# イノベーターのジレンマ

(Clayton Christensen)

- 持続的イノベーション
- 破壊的イノベーション

# 持続的イノベーション

(Sustaining Innovation)

- 今使うのに最適な技術の改善
- 今までよりすぐれた性能を提供
- 現在のマーケットでシェアや経験をもっている企業や大企業が有利
- 問題点「イノベーションのジレンマ」
  - 優良顧客の声だけを聞いて、新しい消費者にフォーカスしそこなう(新参者に未開拓市場を奪われる)
  - 過剰品質・機能・性能を追求し、別の観点からの競争に出遅れてしまう

# 破壊的イノベーション

(Disruptive Innovation)

- これから伸びてくる技術
- 市場を塗り替えてしまう技術革新
- 別の次元で勝負
- 新しい市場の創造
- 新しいユーザーの獲得
- 異分野からの新規参入企業やベンチャー企業が有利(「しがらみ」がない)

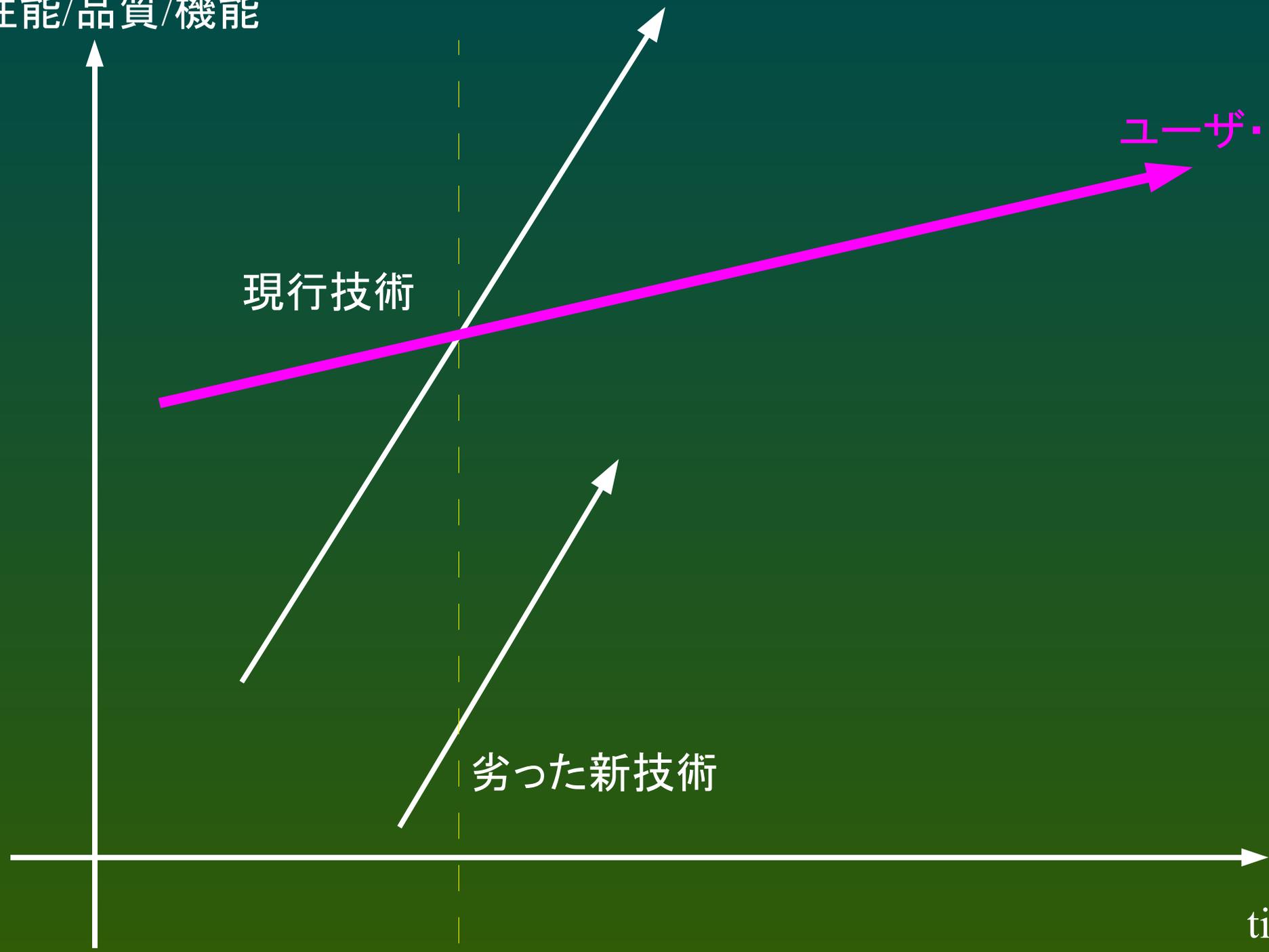
性能/品質/機能

ユーザ・ニーズ

現行技術

劣った新技術

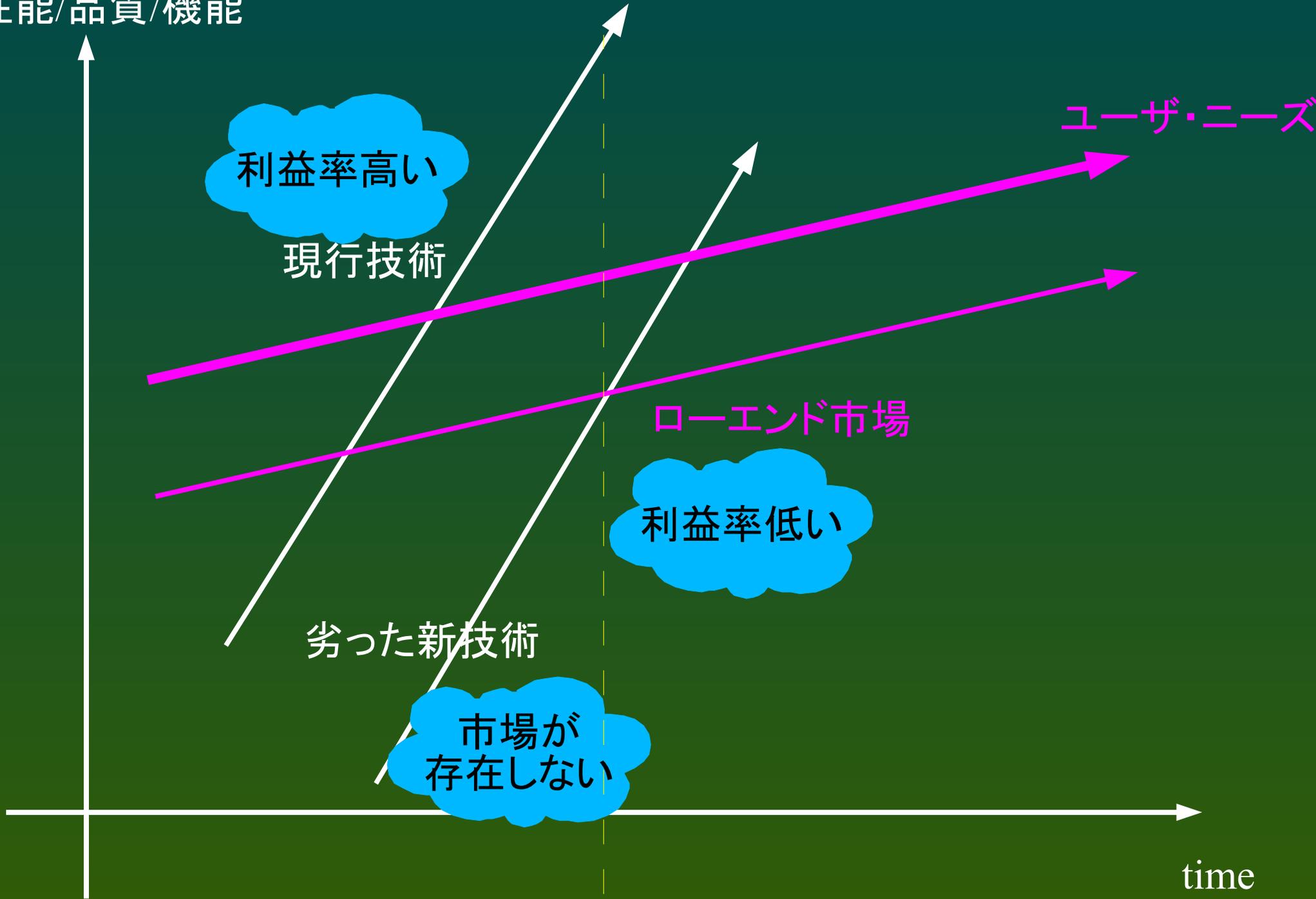
time



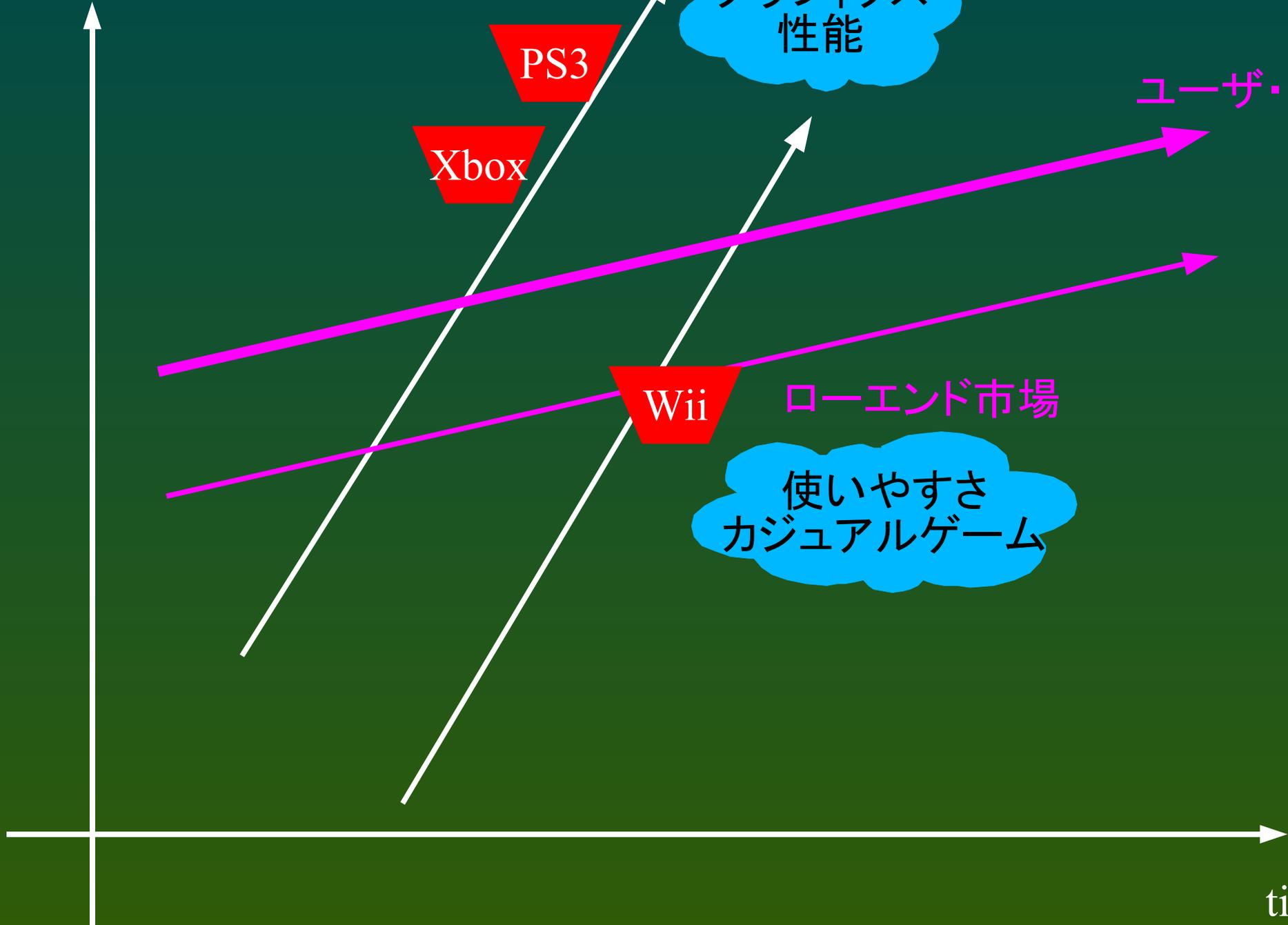
# 破壊的イノベーションの事例

- マイクロプロセッサ(電卓の部品、1971)
- PC(8bit CPUのマニア向けゲーム機、1975?)
- ケータイ(ショルダーフォン、3kg、2万円/月、1985)
- 液晶TV(1.2型、1982、3インチ、9万画素、1987)
- MP3(CDより劣った音質、1991)
- Linux(大学生の学習用OS、1991)
- デジカメ(25万画素、1.8型液晶、1995)
- Ruby(1993)、Java(1995)、PalmPilot(1996)、  
Skype(2004)、...

性能/品質/機能



グラフィクス性能



グラフィクス  
性能

PS3

Xbox

Wii

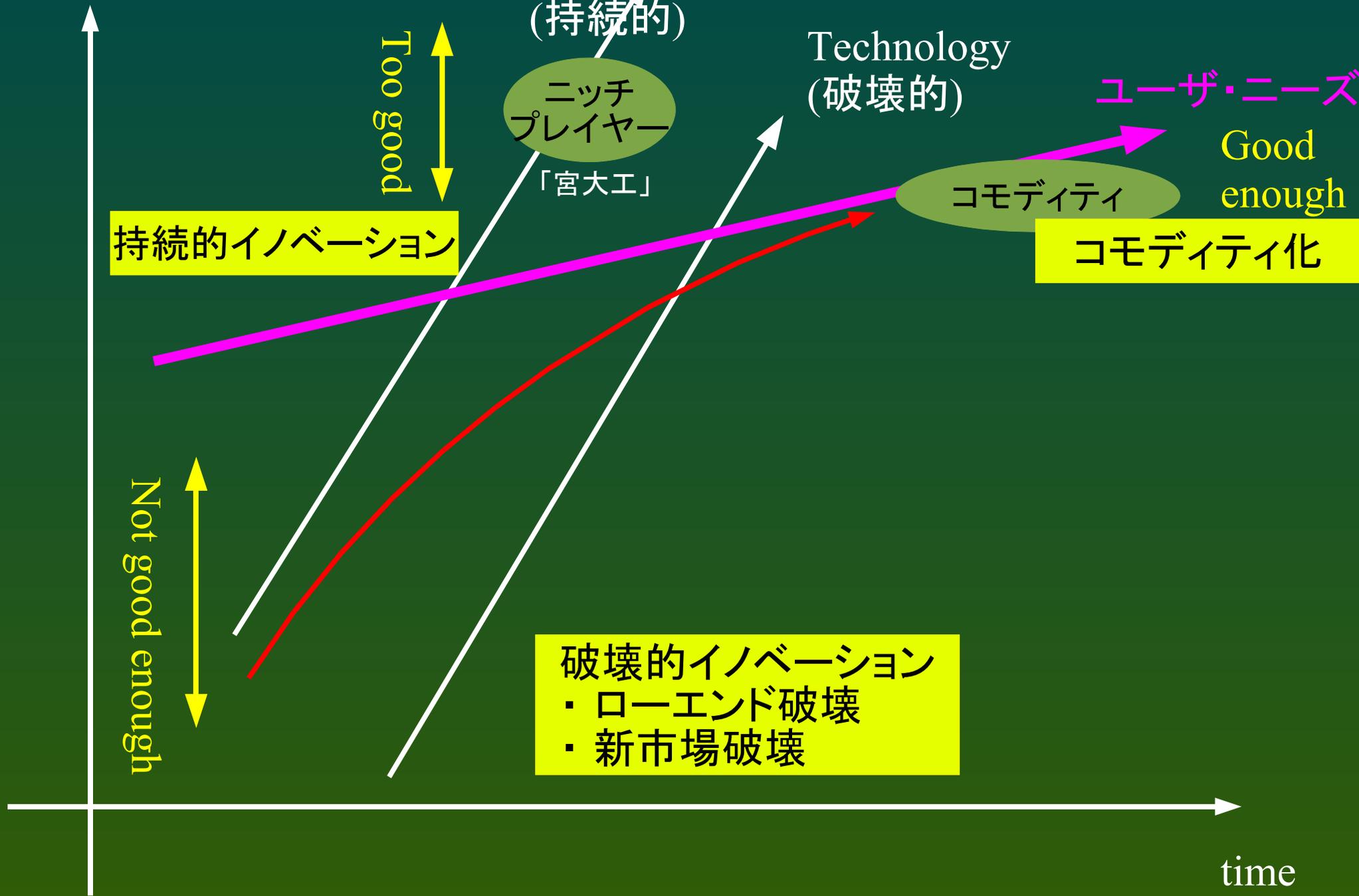
ユーザ・ニーズ

ローエンド市場

使いやすさ  
カジュアルゲーム

time

性能/品質/機能



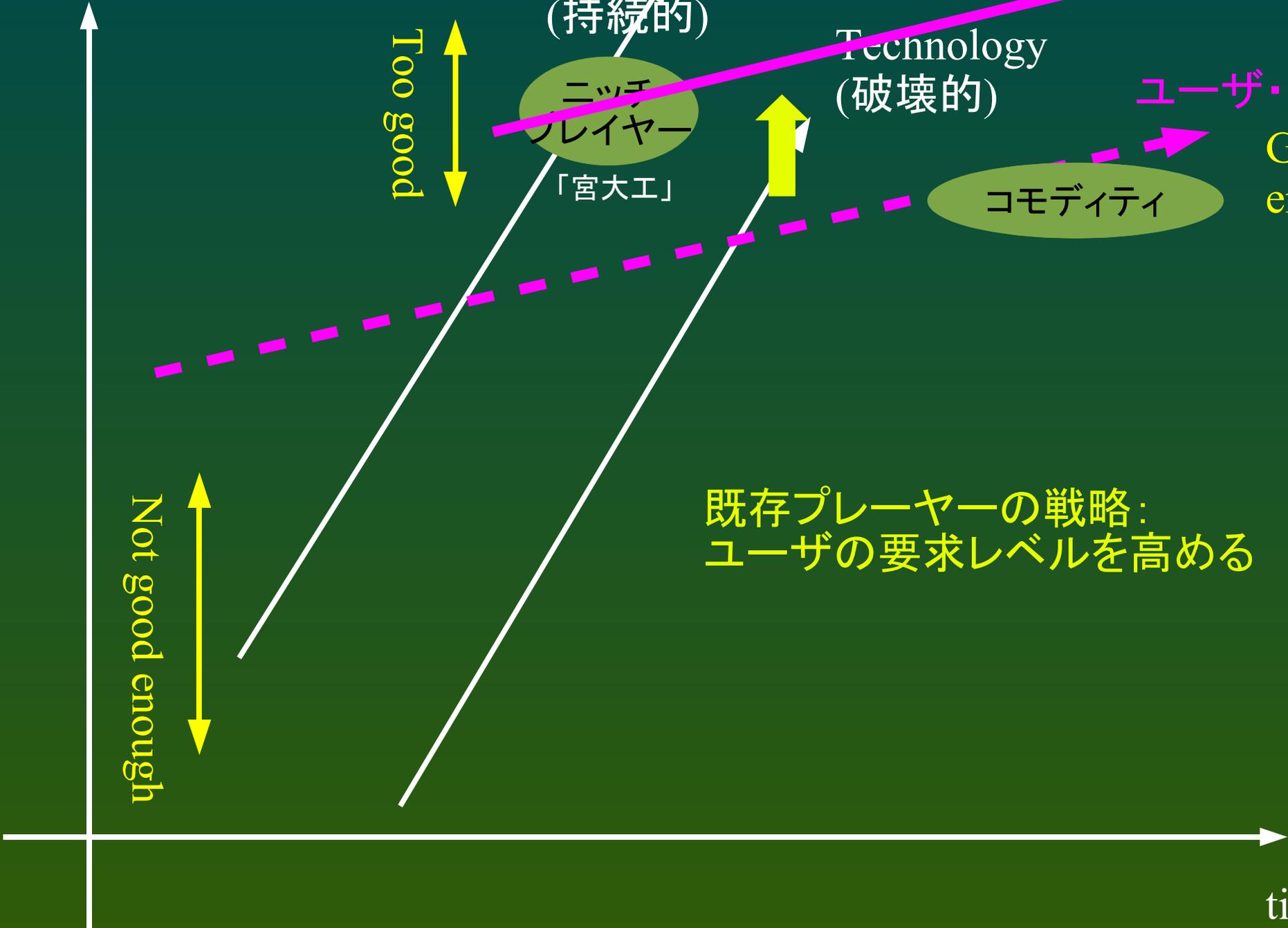
破壊的イノベーション  
・ ローエンド破壊  
・ 新市場破壊

持続的イノベーション

コモディティ化

time

性能/品質/機能



Technology (持続的)

Technology (破壊的)

ニッチプレイヤー  
「宮大工」

コモディティ

ユーザー・ニーズ

Good enough

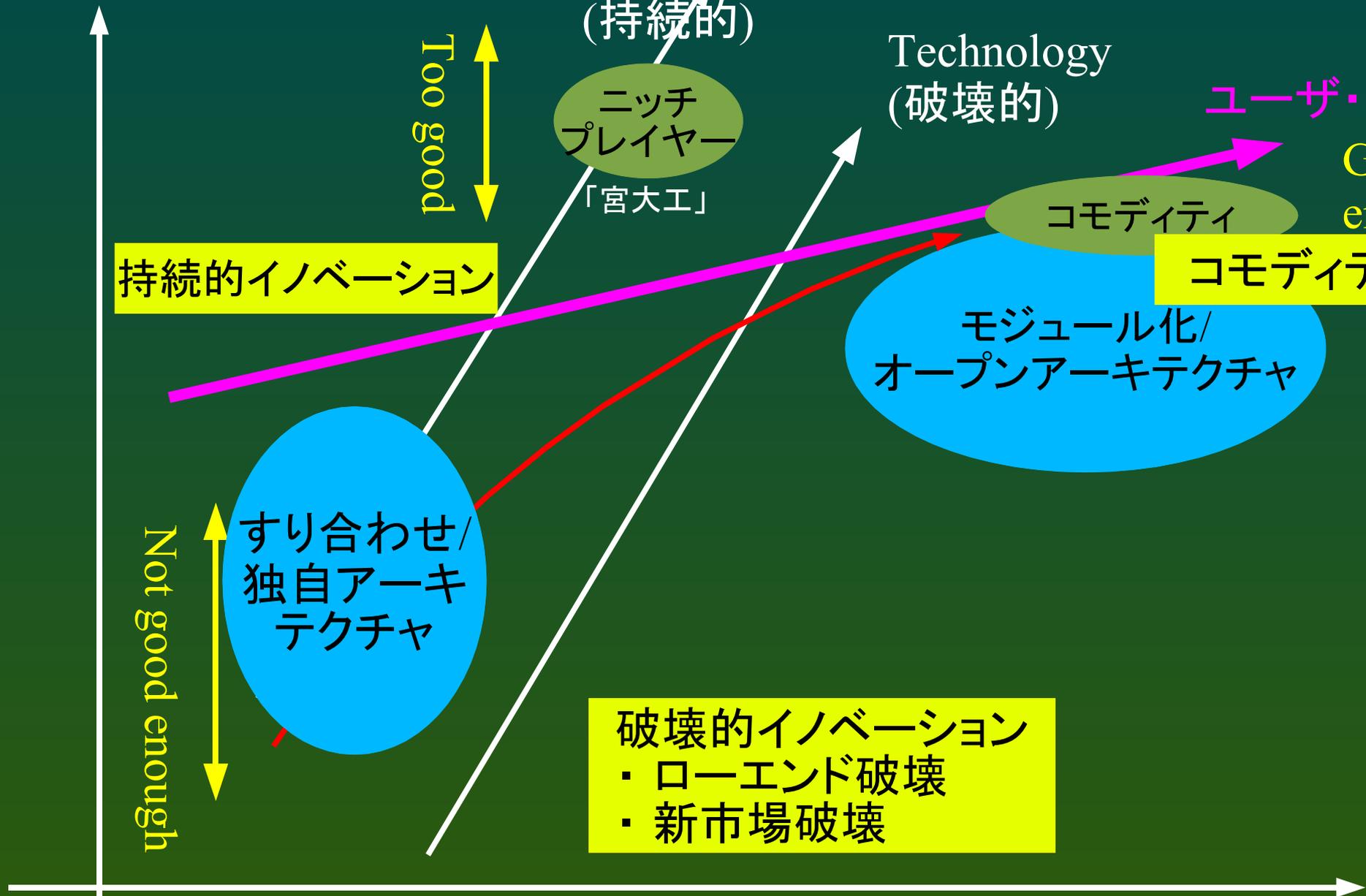
既存プレイヤーの戦略:  
ユーザの要求レベルを高める

Not good enough

Too good

time

性能/品質/機能



持続的イノベーション

Not good enough

すり合わせ/  
独自アーキ  
テクチャ

破壊的イノベーション

- ・ ローエンド破壊
- ・ 新市場破壊

Technology  
(持続的)

ニッチ  
プレイヤー  
「宮大工」

Technology  
(破壊的)

モジュール化/  
オープンアーキテクチャ

コモディティ

コモディティ化

ユーザー・ニーズ

Good  
enough

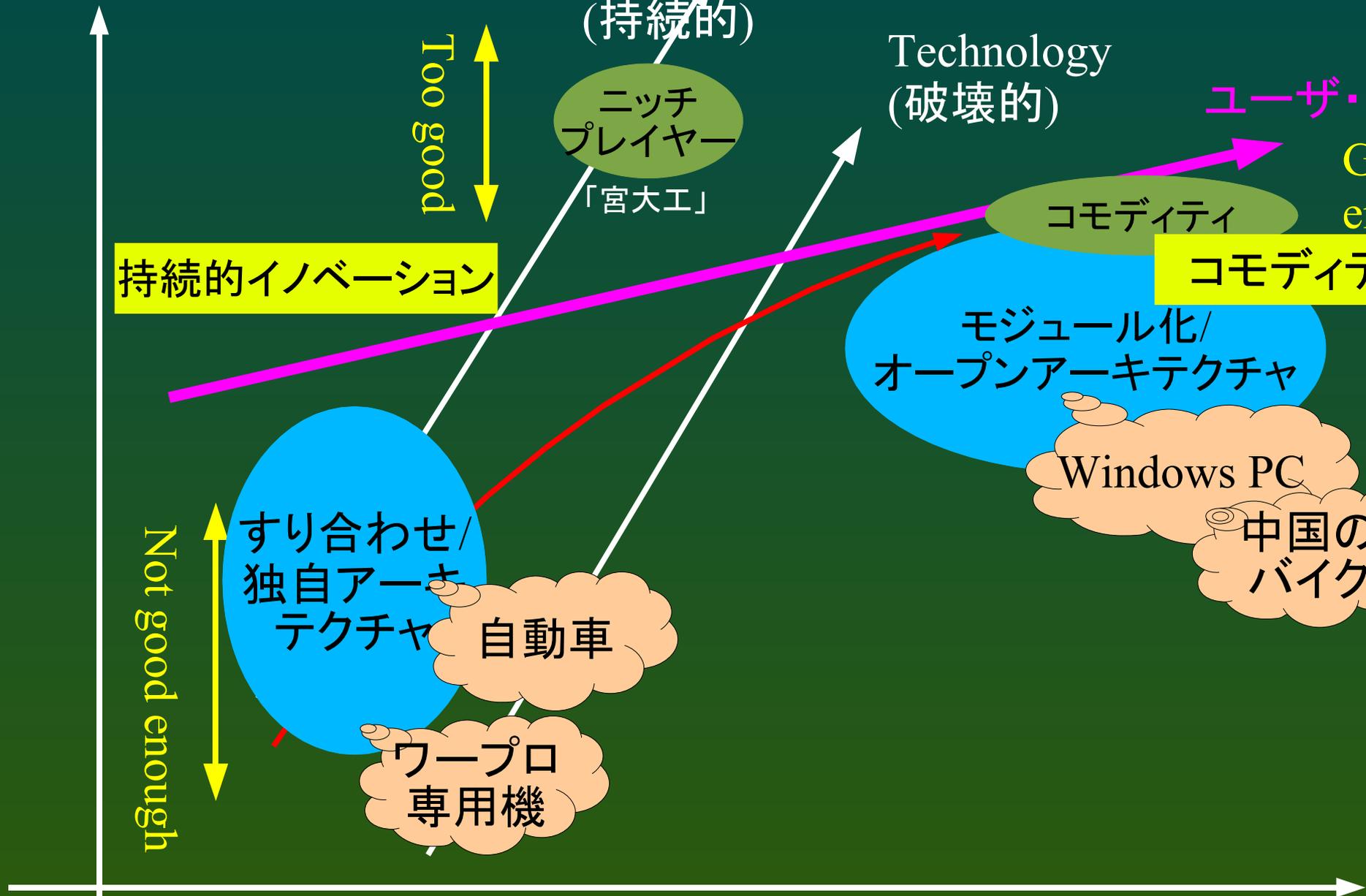
early adoptor  
(bowling lane)

early majority  
(tornado)

late majority  
= cash cow  
(main streets)

time

性能/品質/機能



Technology (持続的)

Technology (破壊的)

ユーザ・ニーズ

Good enough

持続的イノベーション

コモディティ化

モジュール化/  
オープンアーキテクチャ

Windows PC

中国の  
バイク

すり合わせ/  
独自アーキ  
テクチャ

自動車

ワープロ  
専用機

Too good

Not good enough

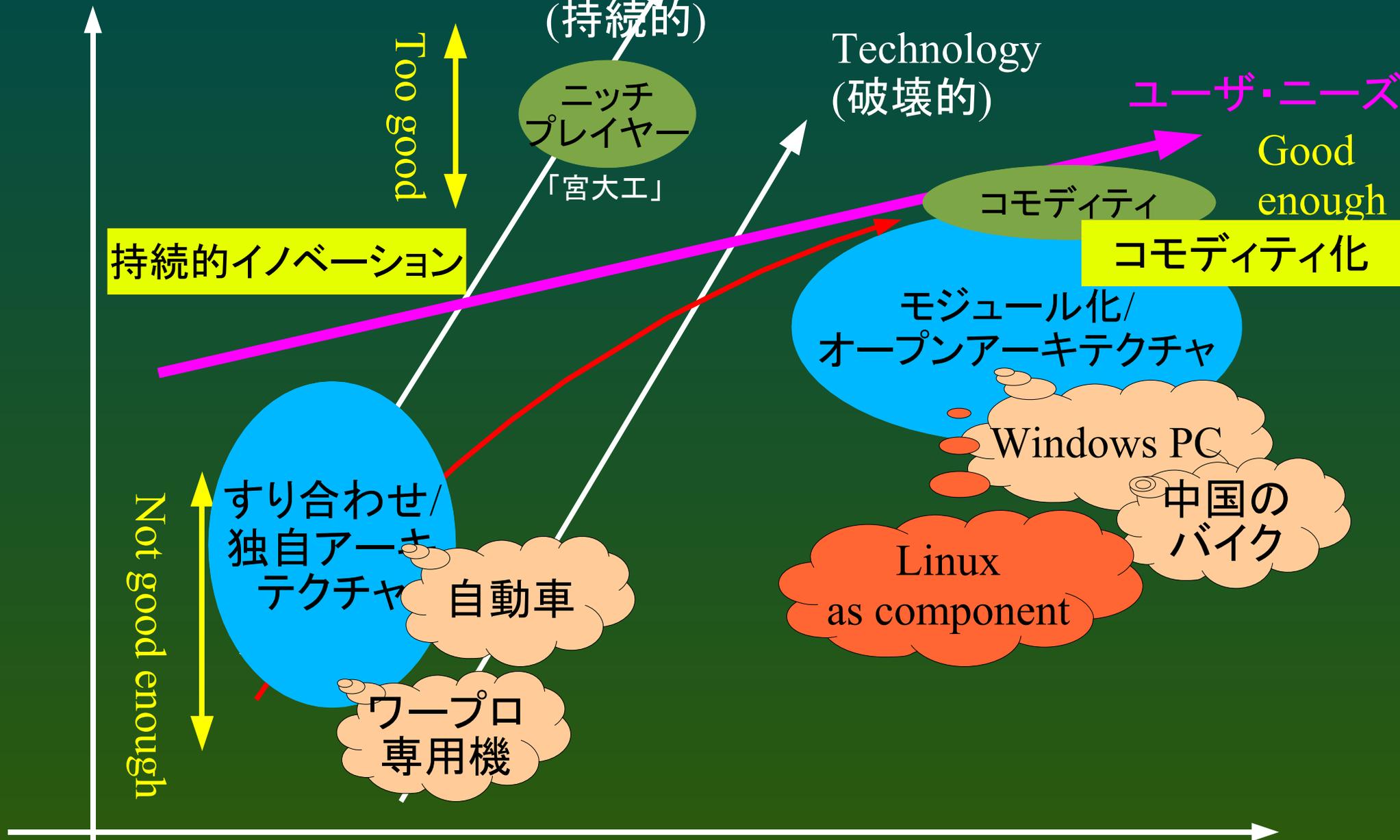
early adoptor  
(bowling lane)

early majority  
(tornado)

late majority  
= cash cow  
(main streets)

time

性能/品質/機能



持続的イノベーション

Too good

Not good enough

すり合わせ/  
独自アーキ  
テクチャ

自動車

ワープロ  
専用機

Technology  
(持続的)

ニッチ  
プレイヤー  
「宮大工」

Technology  
(破壊的)

モジュール化/  
オープンアーキテクチャ

Windows PC

Linux  
as component

中国の  
バイク

コモディティ

コモディティ化

ユーザ・ニーズ

Good  
enough

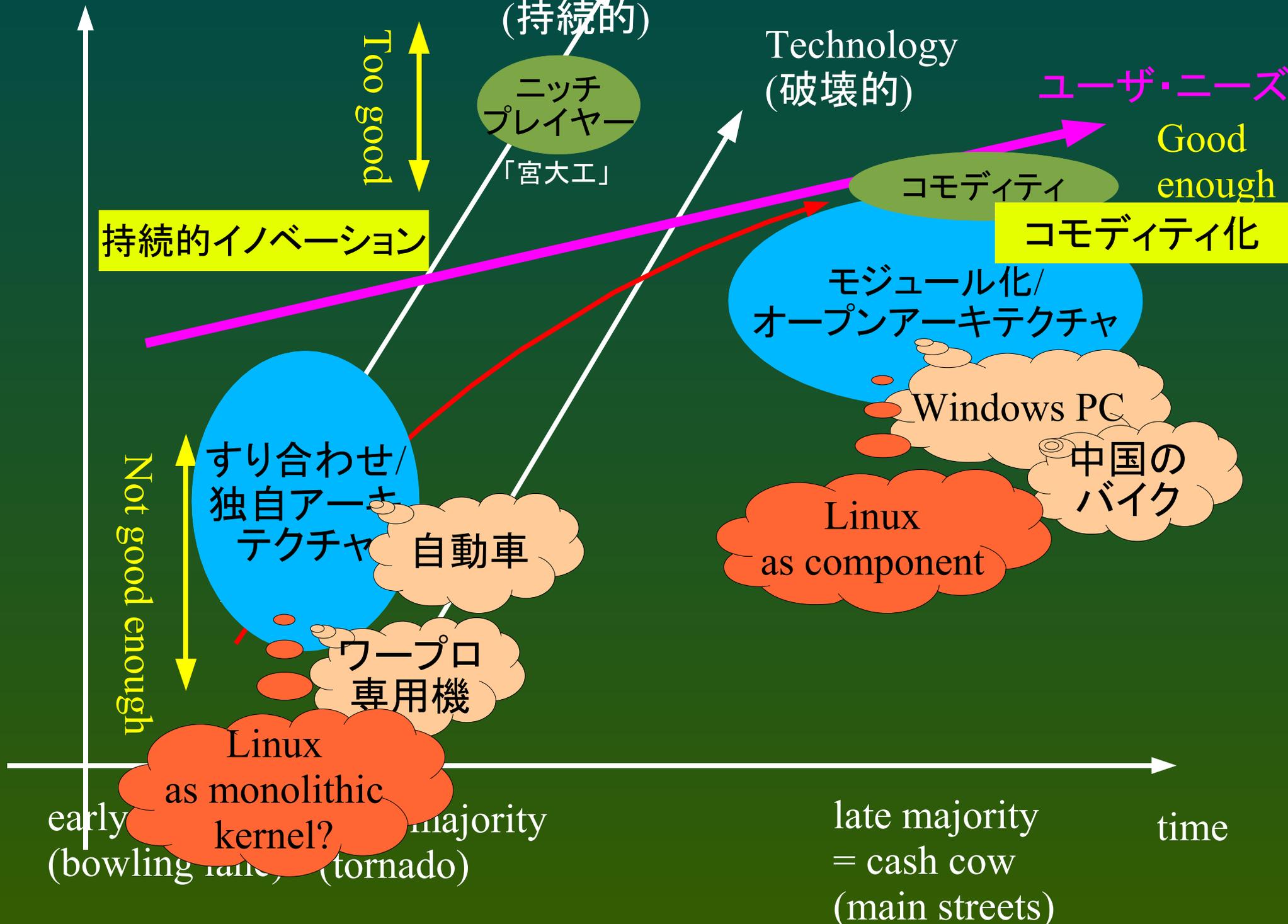
early adoptor  
(bowling lane)

early majority  
(tornado)

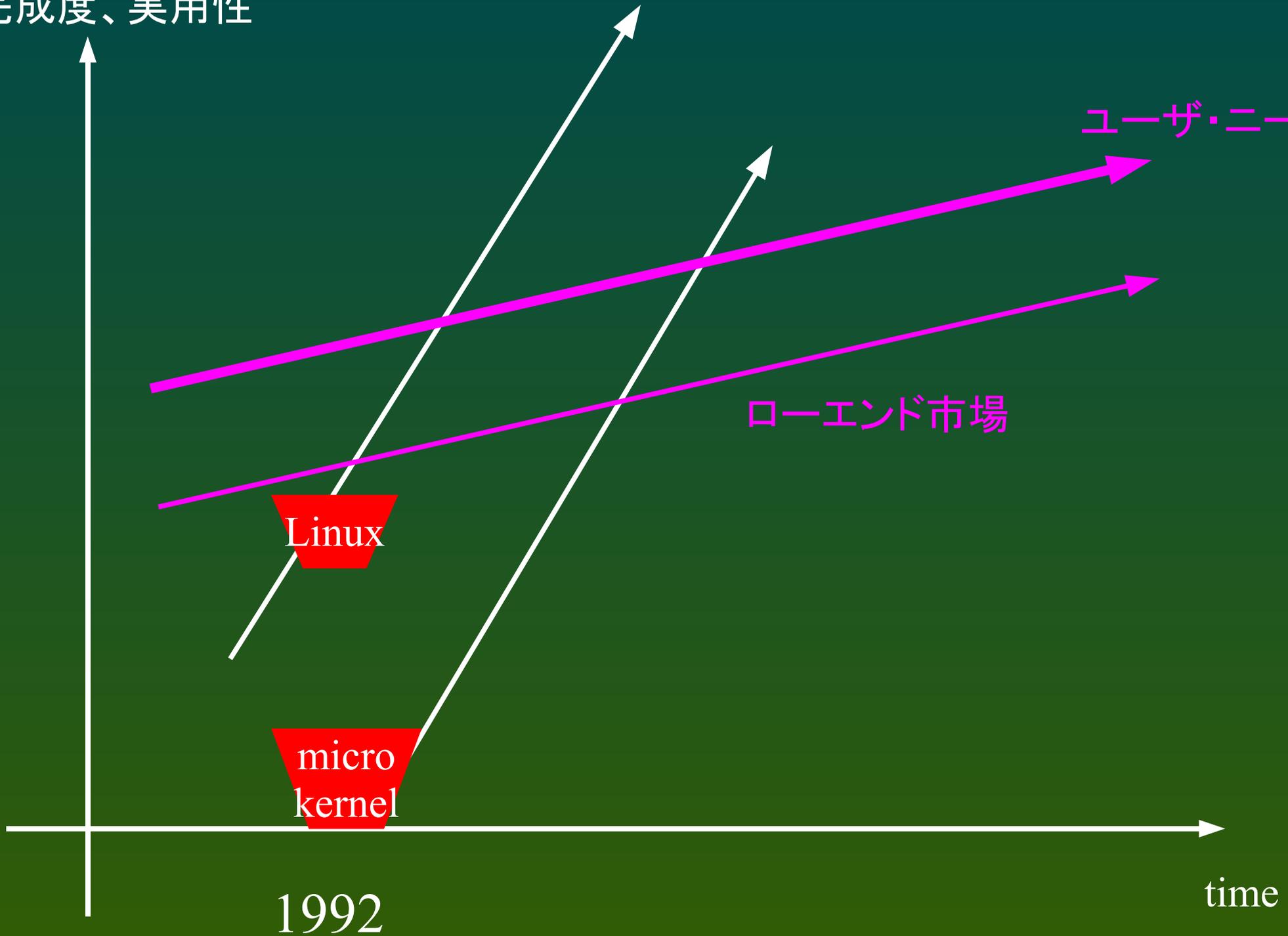
late majority  
= cash cow  
(main streets)

time

性能/品質/機能



完成度、実用性



Linux

micro  
kernel

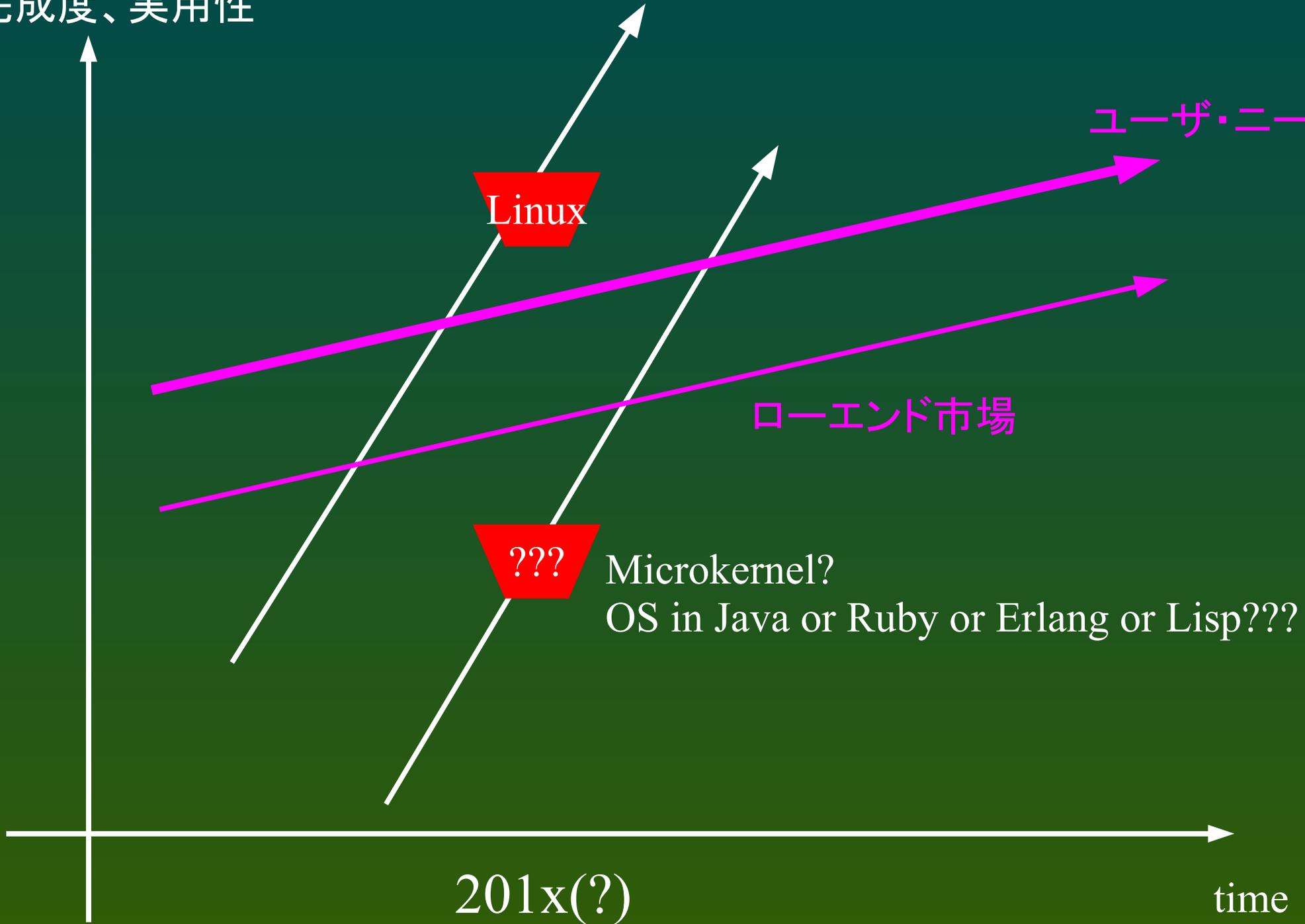
ローエンド市場

ユーザ・ニーズ

1992

time

完成度、実用性



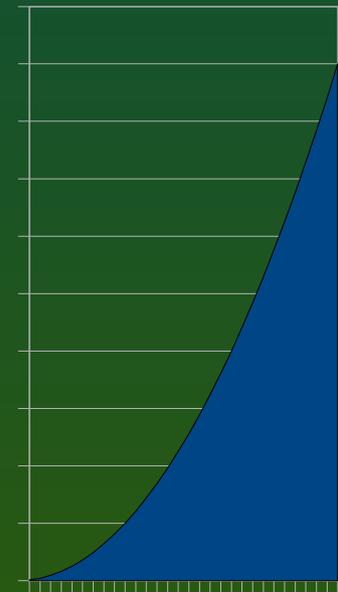
# 組込みソフトはいつまでCで書くのか？

- 私が新入社員だった頃(いつだ?)
  - みんなアセンブリ言語でプログラミングしていた
    - オフコンのシステムプログラム、ワープロ専用機、電子手帳
  - C言語やPascalを使おうとプロモートした
    - 「遅い！」「メモリを食いすぎる！」「使い物にならん！」
    - 「既存のコードを逆コンパイルしてくれ」
- 現在
  - 高速化、省メモリのためにC言語で書いている
  - でも、ソフトウェアの大規模化により、複雑さに対処しきれなくなってきた
- 「本物のプログラマは\*\*\*を使わない!？」

# 複雑化への対処

- 分割統治(divide and conquer)
  - モジュール化、フレームワーク化
- 大粒度化
  - アセンブリ言語→C/C++→???

複雑さ



規模

# 破壊的イノベーションの芽を みつけるためのキーワード

## • こんなもの

- 遅くて/ややこしくて/重すぎて/高すぎて/難しすぎて、  
使いものにならない
- 商品のクォリティでない
- 安物、まがいもの、おもちゃ
- 趣味で使うならいいけど。マニア向け
- きれいごと。世間じゃ通用しない。わりきりすぎ

# Tim O'reillyの有望技術発見基準

<http://www.oreillyn.net.com/pub/wlg/3123>

- hackability
  - その道のプロをにやりとさせるものがある
- being in line with some major trend
  - 追い風が吹いている
- disruptive potential
  - なにかをぶっとばしかねない迫力がある
- grassroots enthusiasm
  - 熱気にみちたひとびとの支持がある
- the presence of professional practitioners
  - 仕事をきちんとこなすプロがいる
- a possible business ecology
  - 商売になりそう、ビジネス生態系が成立しうる

**SHARP**