Your org as an API for platform adoption

#APIStrat

@rchoi

@rchoi

@rchoi

- Early engineer, salesforce.com

@rchoi

- Early engineer, salesforce.com
- Director of Engineering, Zuora

@rchoi

- Early engineer, salesforce.com
- Director of Engineering, Zuora
- MBA, MIT Sloan

@rchoi

- Early engineer, salesforce.com
- Director of Engineering, Zuora
- MBA, MIT Sloan
- Head of Partner Engineering, Twitter

Org as an API for platform adoption

Org as an API for platform adoption

- Platform strategy

Org as an API for platform adoption

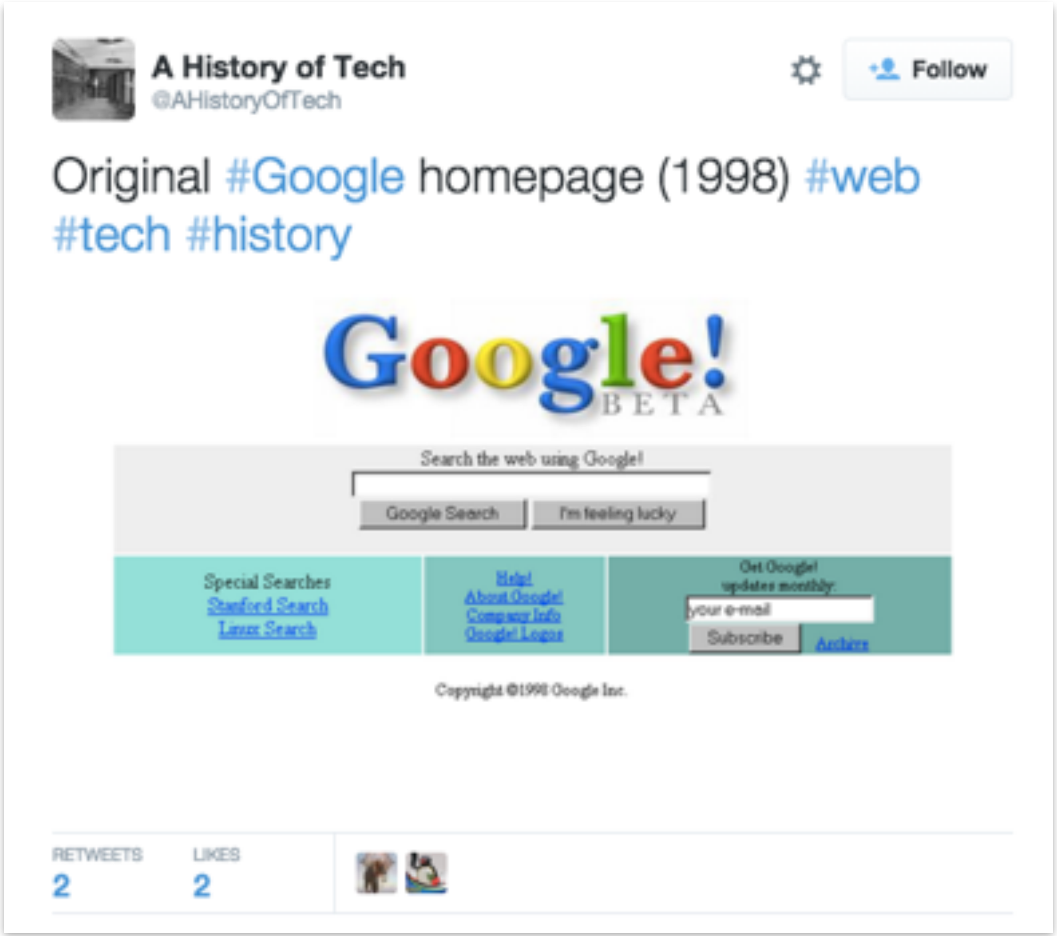
- Platform strategy
- Organizational structures and networks

Org as an API for platform adoption

- Platform strategy
- Organizational structures and networks
- Organization design

Platform strategy

- Technology-push vs. Market-pull



A History of Tech
@AHistoryOfTech

Follow

Original #Google homepage (1998) #web #tech #history

Google! BETA

Search the web using Google!

Google Search | I'm feeling lucky

Special Searches
Stanford Search
Linux Search

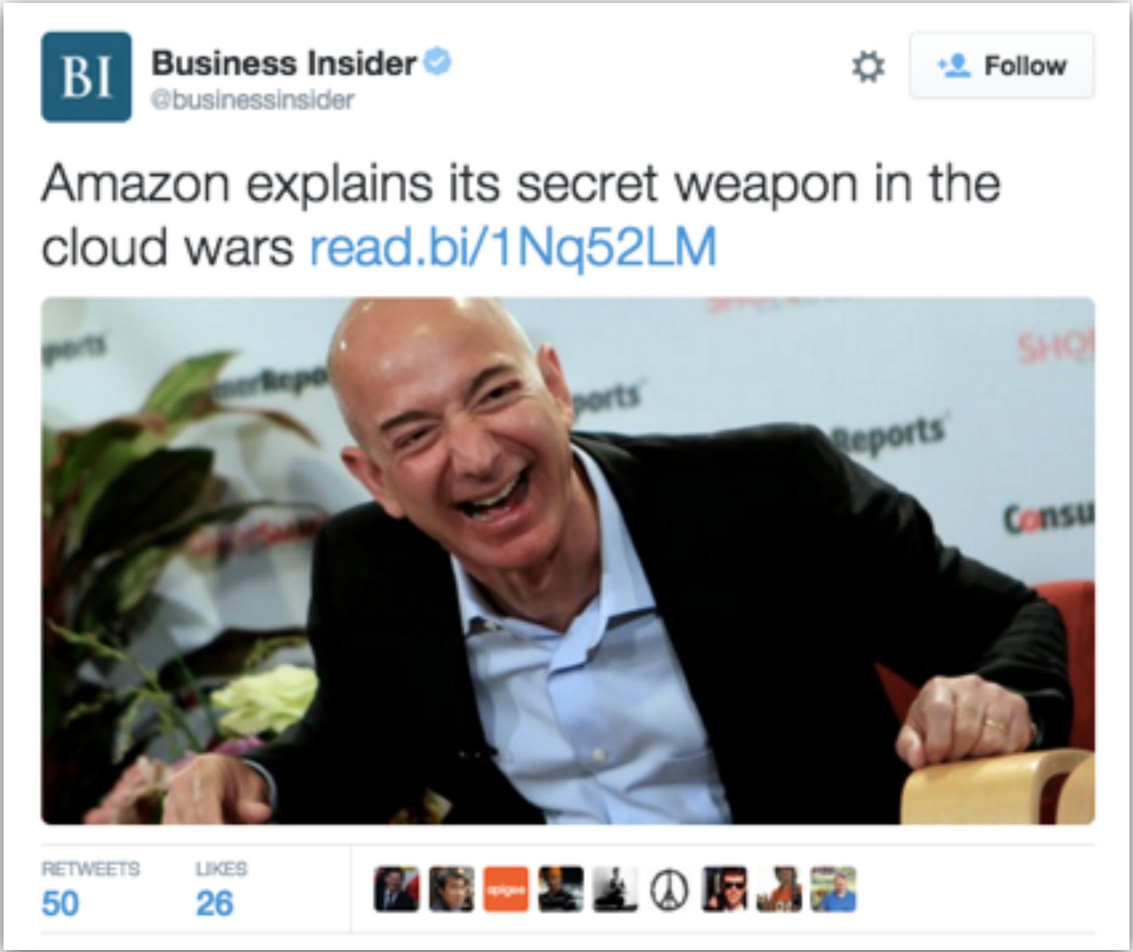
Help!
About Google!
Company Info
Google! Logos

Get Google!
updates monthly:
your e-mail
Subscribe
Archive

Copyright ©1998 Google Inc.

RETWEETS
2

LIKES
2



Business Insider
@businessinsider

Follow

Amazon explains its secret weapon in the cloud wars read.bi/1Nq52LM

RETWEETS 50
LIKES 26

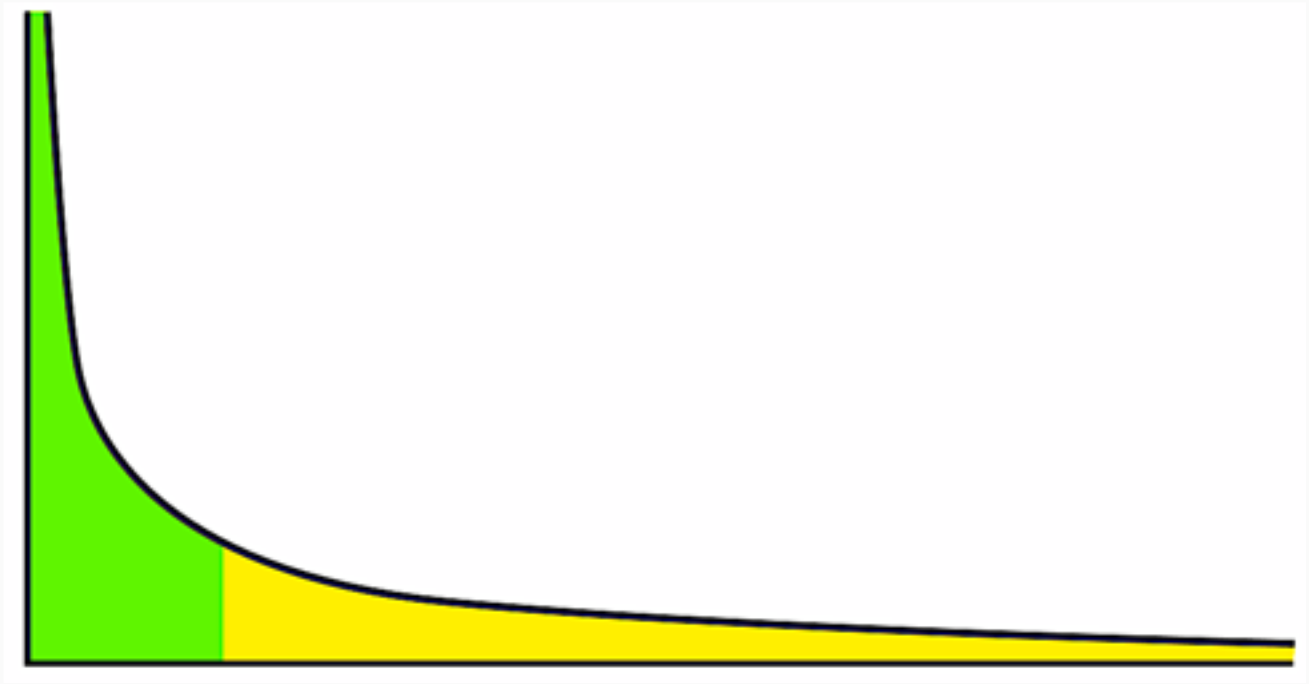
Platform strategy

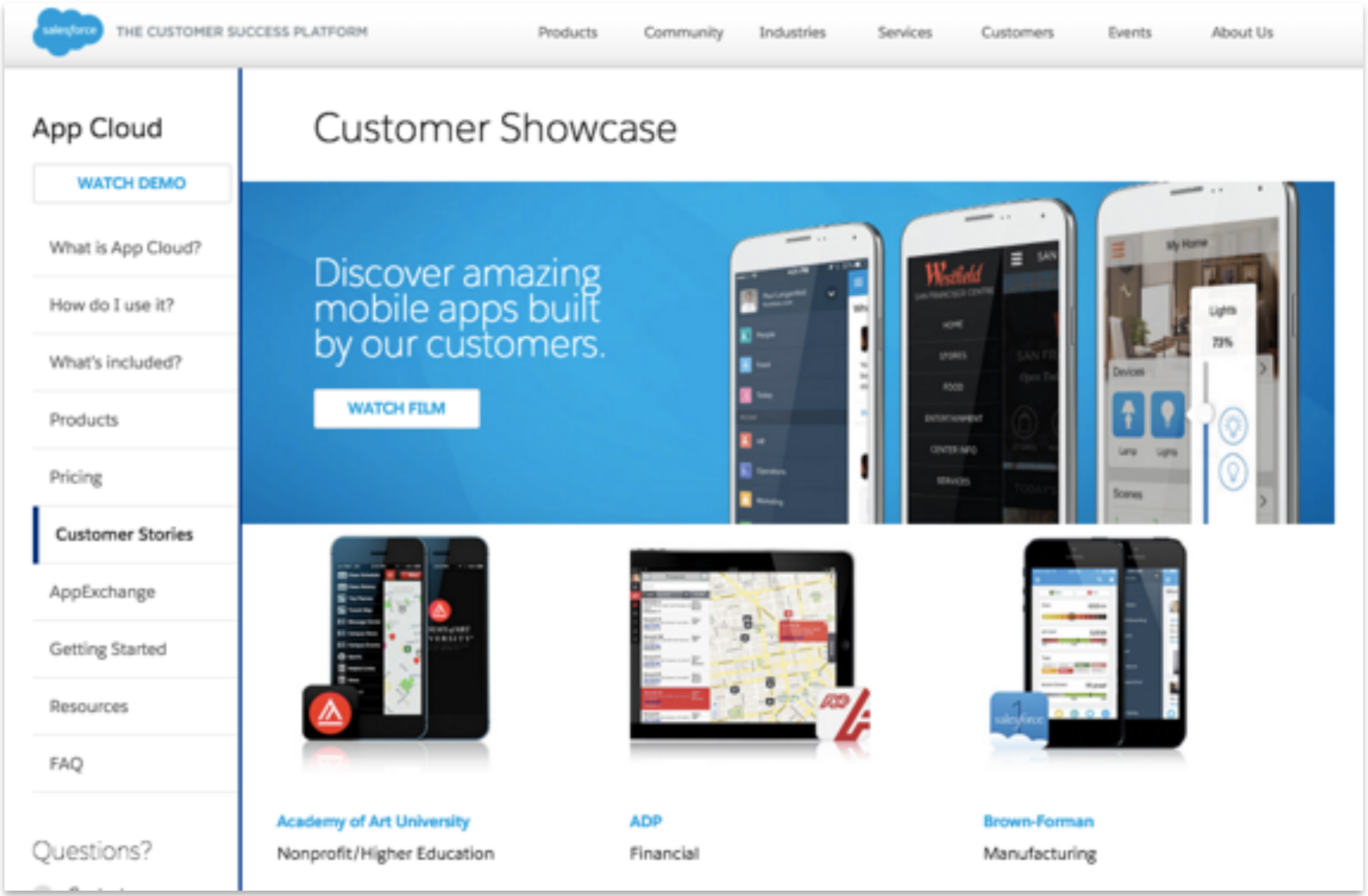
- Technology-push vs. Market-pull
- Stage: Feedback vs. Adoption

Platform strategy

- Technology-push vs. Market-pull
- Stage: Feedback vs. Adoption
- Adoption: Head vs. Tail

Platform strategy





salesforce THE CUSTOMER SUCCESS PLATFORM

Products Community Industries Services Customers Events About Us

App Cloud

WATCH DEMO

- What is App Cloud?
- How do I use it?
- What's included?
- Products
- Pricing
- **Customer Stories**
- AppExchange
- Getting Started
- Resources
- FAQ

Questions?

Customer Showcase

Discover amazing mobile apps built by our customers.

WATCH FILM

Academy of Art University

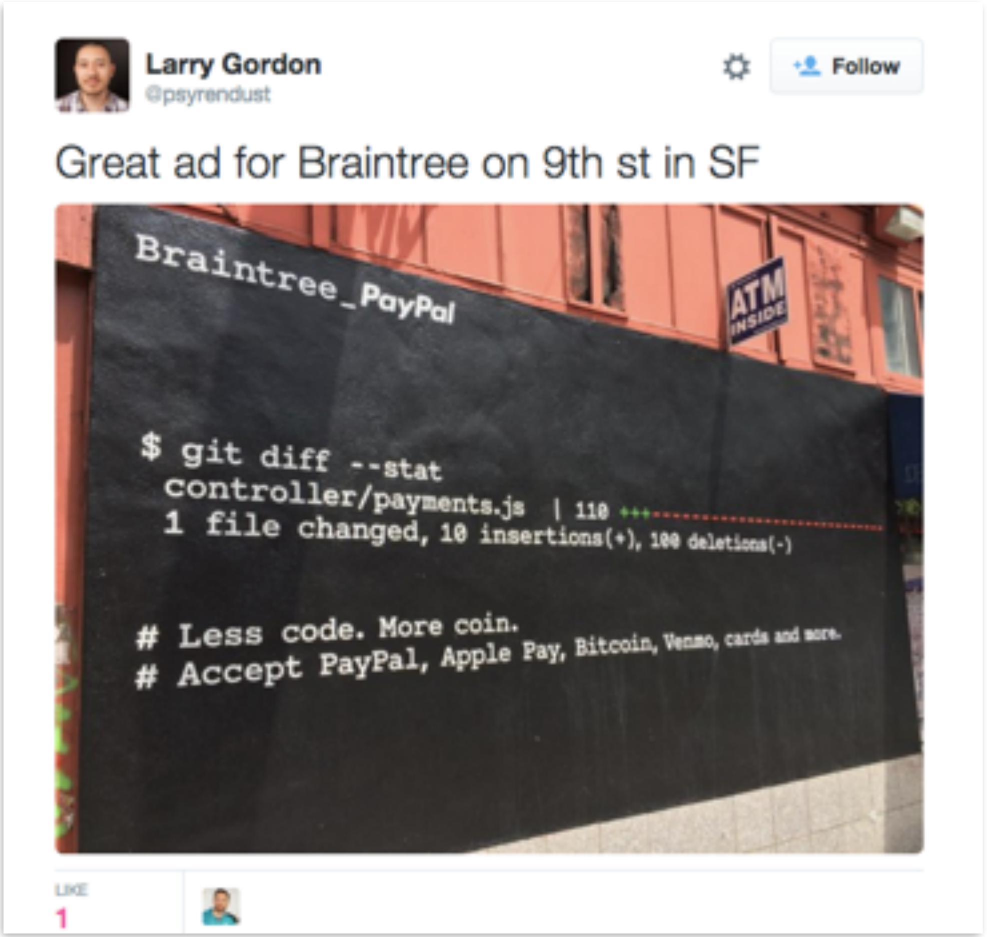
Nonprofit/Higher Education

ADP

Financial

Brown-Forman

Manufacturing



Larry Gordon
@psyrendust

Follow

Great ad for Braintree on 9th st in SF

```
Braintree_PayPal

$ git diff --stat
 controller/payments.js  | 110 +++----------------------------
 1 file changed, 10 insertions(+), 100 deletions(-)


# Less code. More coin.
# Accept PayPal, Apple Pay, Bitcoin, Venmo, cards and more.
```

LIKE
1

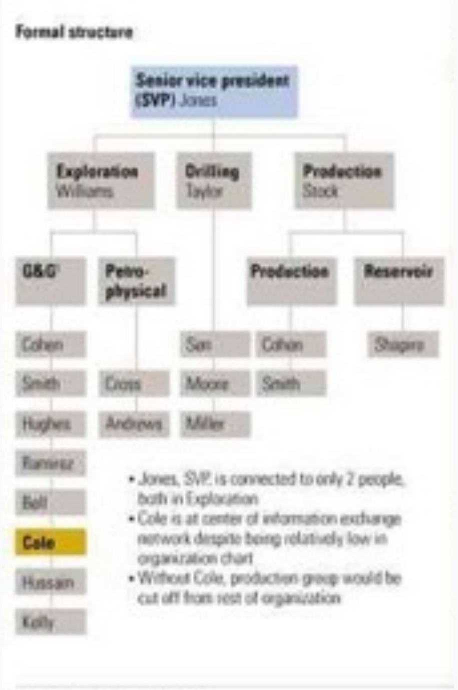
Platform strategy

- Technology-push vs. Market-pull
- Stage: Feedback vs. Adoption
- Adoption: Head vs. Tail

Organizational structure

Informal networks: the company behind the chart (HBR, 1993)

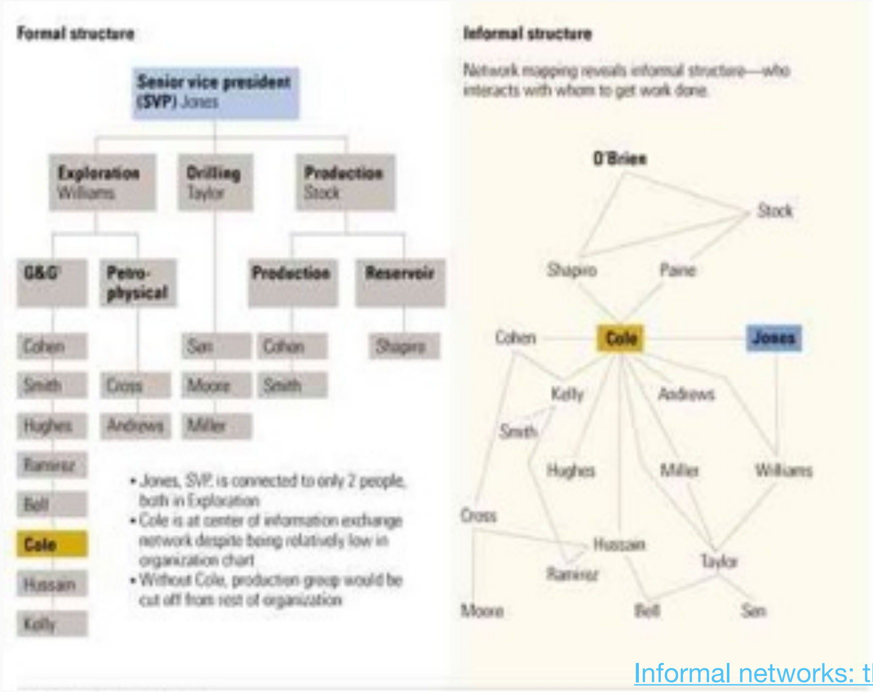
Organizational structure



- Jones, SVP, is connected to only 2 people, both in Exploration
- Cole is at center of information exchange network despite being relatively low in organization chart
- Without Cole, production group would be cut off from rest of organization

Informal networks: the company behind the chart (HBR, 1993)

Organizational networks



- Jones, SVP, is connected to only 2 people, both in Exploration
- Cole is at center of information exchange network despite being relatively low in organization chart
- Without Cole, production group would be cut off from rest of organization

[Informal networks: the company behind the chart (HBR, 1993)]()

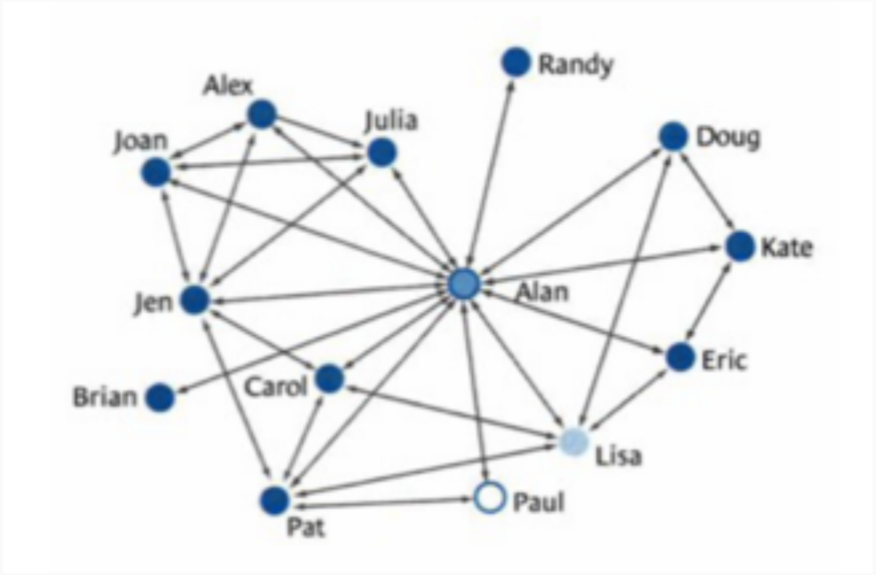
Organizational networks

- Connectors
- Brokers
- Specialists

The role of networks in organizational change (McKinsey, 2007)
Public Human Resource Management (Ryu, 2012)

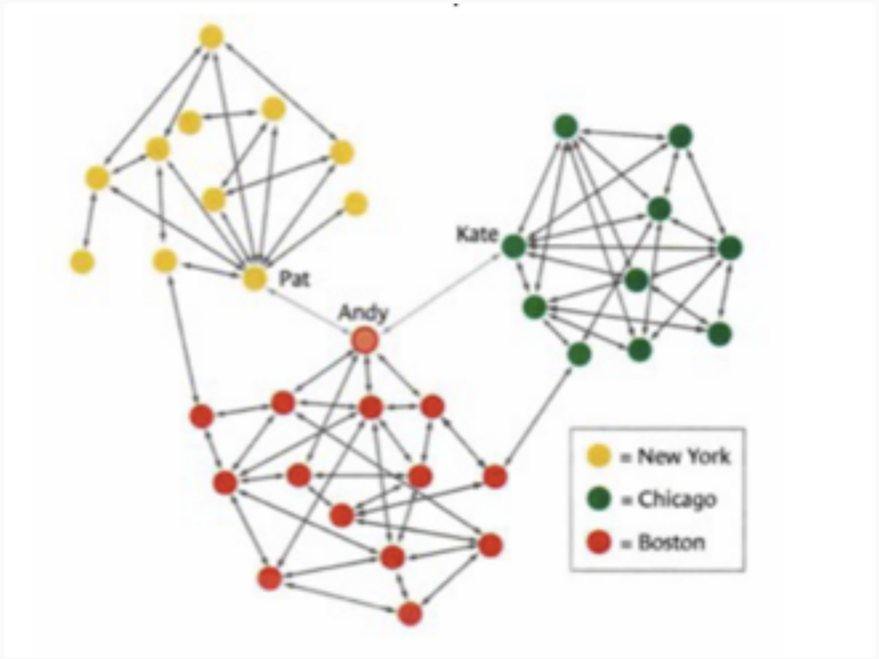
Networks: Connectors

- People whom others consult frequently for information, expertise or decision-making



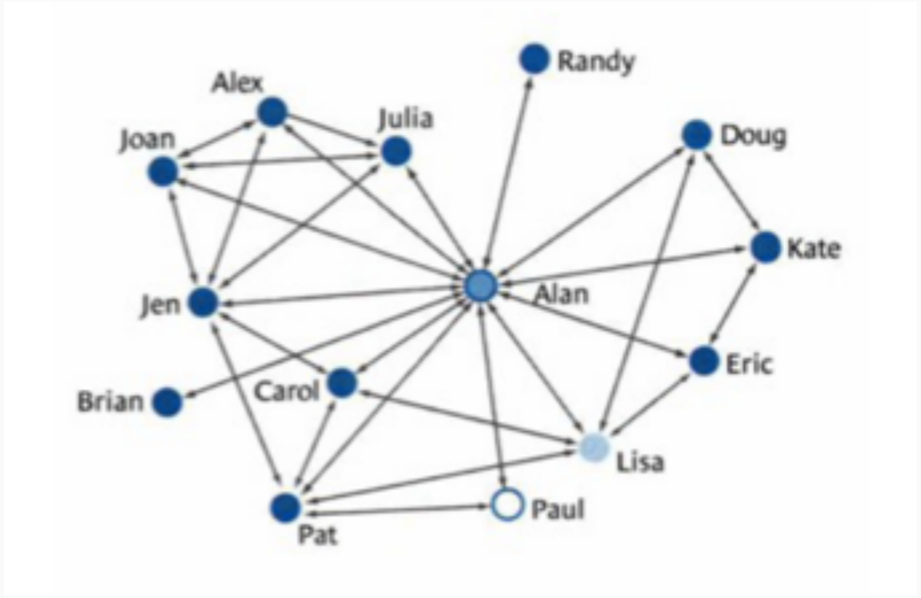
Networks: Brokers

- People who connect different subgroups in network
- Subgroups are often departments, locations, divisions, tenures



Networks: Specialists

- Experts with specific kinds of information or technical knowledge



Organization design

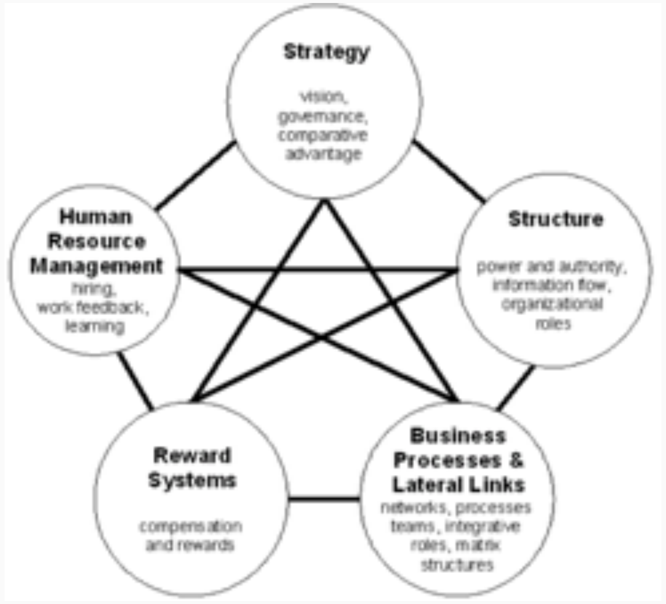
[Designing Organizations (Galbraith, 1995)]

Organization design

“Organization design is the deliberate process of configuring structures, processes, reward systems, and people practices to create an effective organization capable of achieving the business strategy.”

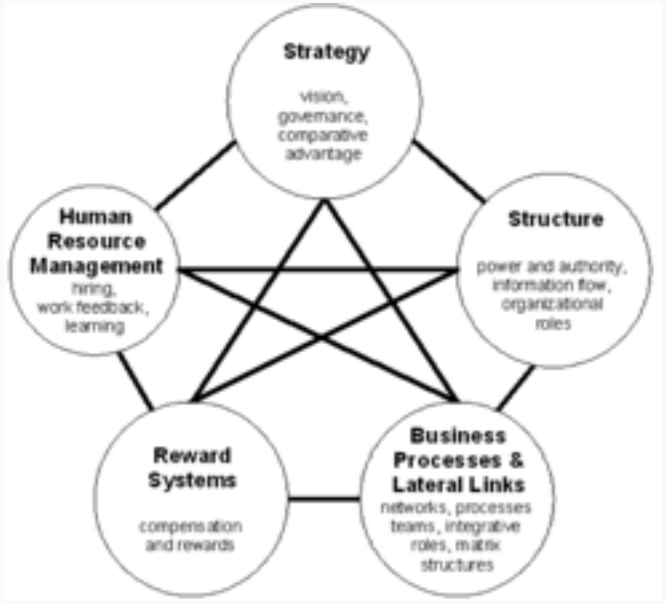
Designing Organizations (Galbraith, 1995)

Organization design: Star model


Strategy
vision, governance, comparative advantage
Structure
power and authority, information flow, organizational roles
Business Processes & Lateral Links
networks, processes teams, integrative roles, matrix structures
Reward Systems
compensation and rewards
Human Resource Management
hiring, work feedback, learning

Designing Organizations (Galbraith, 1995)

Organization design: Star model


Strategy
vision, governance, comparative advantage
Structure
power and authority, information flow, organizational roles
Business Processes & Lateral Links
networks, processes teams, integrative roles, matrix structures
Reward Systems
compensation and rewards
Human Resource Management
hiring, work feedback, learning

- **Strategy**
- **Structure**
- **Resources**
- Rewards
- Processes

Designing Organizations (Galbraith, 1995)

Org design: Departments

```
class Department():

   def __init__(self, strategy=None, structure=None, resources=None, rewards=None, processes=None):

        self.strategy = strategy
        self.resources = resources
        self.structure = structure
#         self.rewards = rewards
#         self.processes = processes

    def execute(self, departments):

        return None
```

Org design: Product

```
class Product(Department):

    def __init__(self):

        self.strategy = "Capture of opportunities via product development & innovation"
        self.resources = "Product development”
        self.structure = "Specialist"
```

Org design: Product

```
class Product(Department):

    def __init__(self):

        self.strategy = "Capture of opportunities via product development & innovation"
        self.resources = "Product development"
        self.structure = "Specialist"

    def execute(self, departments):

        if Marketing in departments:
            return "Product optimized for long-term market”

        if Sales in departments or BusinessDevelopment in departments:
            return "Product optimized for short-term wins”
```

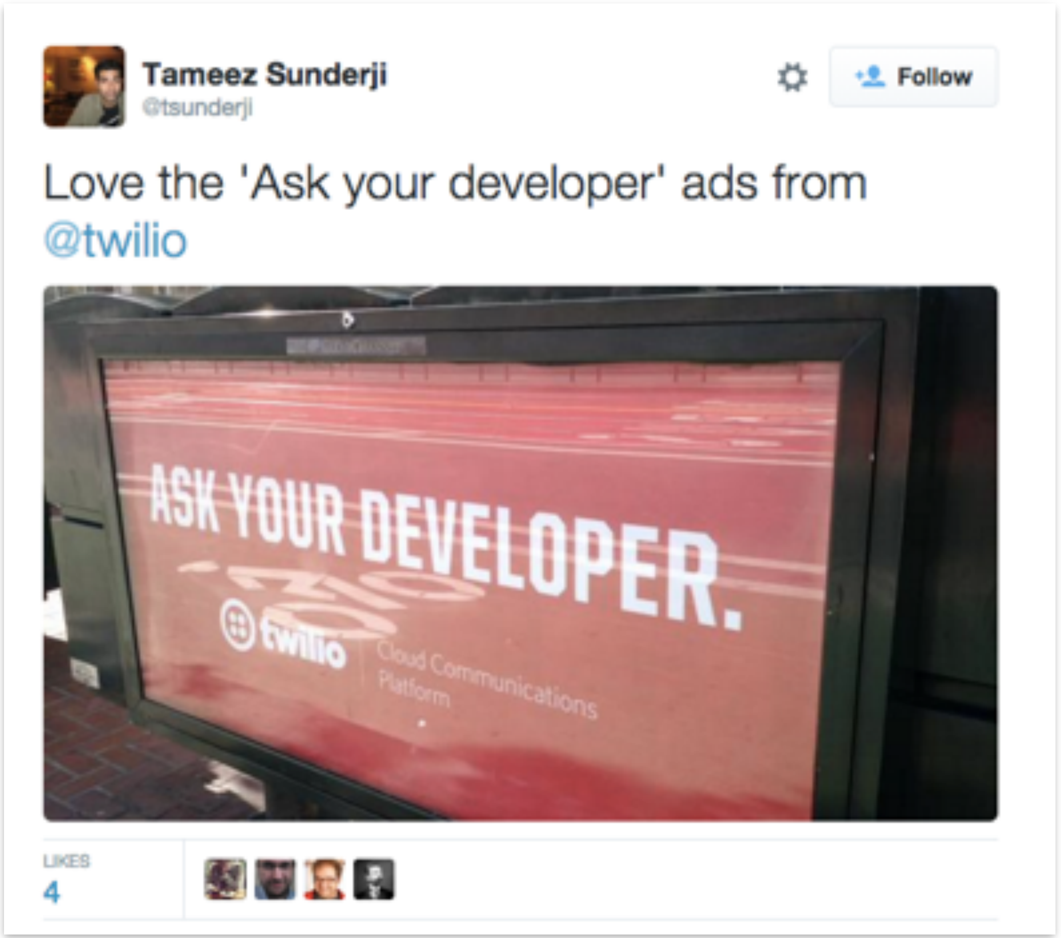
Org design: Marketing

Org design: Marketing

```
class Marketing(Department):

    def __init__(self):

        self.strategy = "Discovery and creation of global optima opportunities"
        self.resources = ["Market research & sizing", "Collateral”,
                          "Lead pipeline generation”]
        self.structure = "Connector"
```

Org design: Marketing

```
class Marketing(Department):

    def __init__(self):

        self.strategy = "Discovery and creation of global optima opportunities"
        self.resources = ["Market research & sizing", "Collateral”,
                          "Lead pipeline generation”]
        self.structure = "Connector"

    def execute(self, departments=None):

        return “Go-to-market strategy"
```

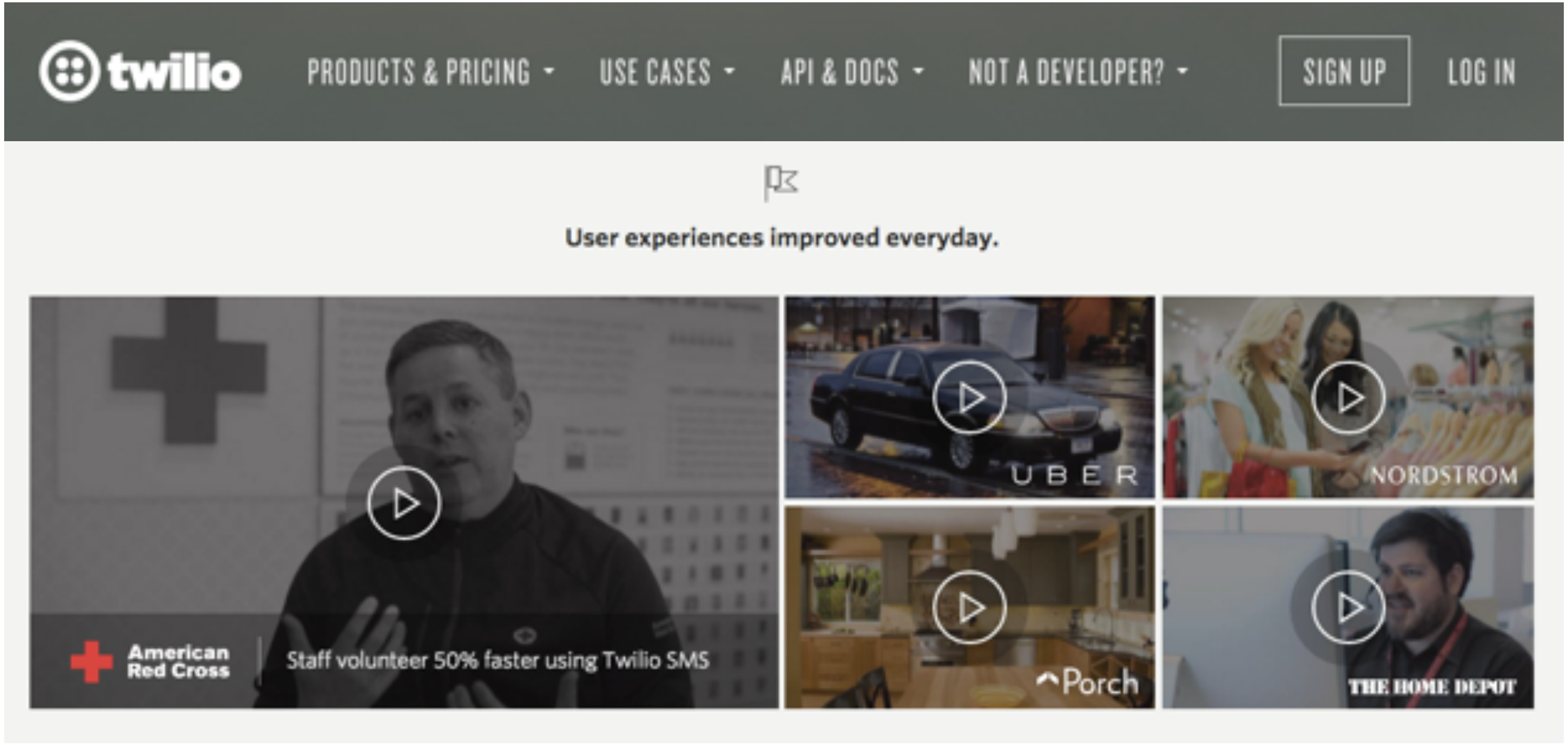


Tameez Sunderji
@tsunderji

Follow

Love the 'Ask your developer' ads from @twilio

LIKES
4



User experiences improved everyday.

Staff volunteer 50% faster using Twilio SMS

Org design: Sales

```
class Sales(Department):

    def __init__(self):

        self.strategy = “Growth via direct customer engagement”
        self.resources = ["Prospecting & qualification", “Negotiation"]
        self.structure = "Specialist"
```

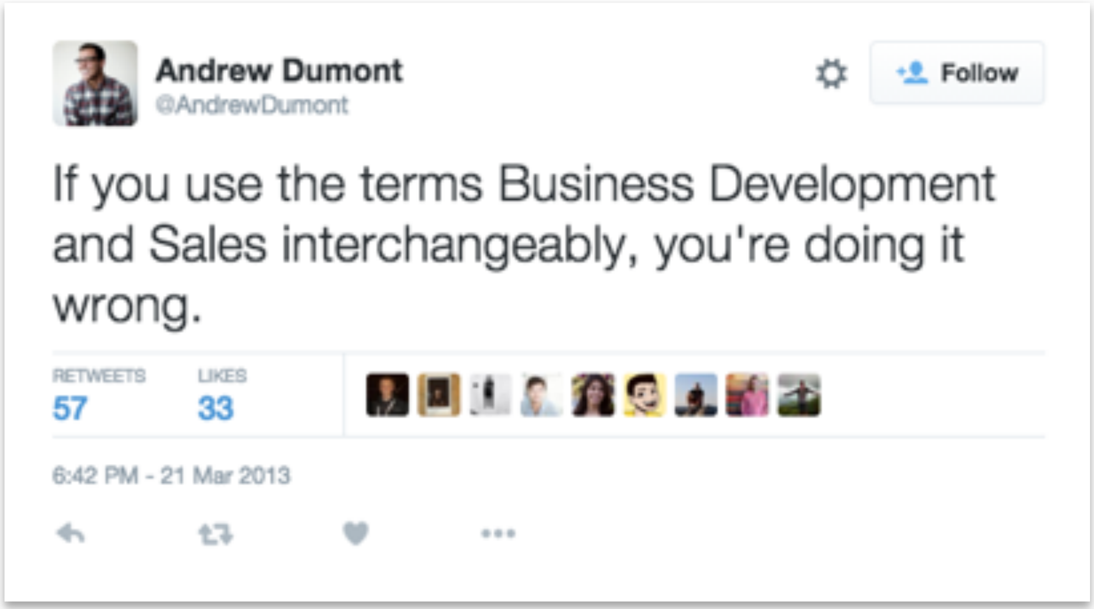
Org design: Sales

```
class Sales(Department):

    def __init__(self):

        self.strategy = “Growth via direct customer engagement”
        self.resources = ["Prospecting & qualification", “Negotiation"]
        self.structure = "Specialist"

    def execute(self, departments):

        return “$$$”
```

Org design: Sales

```python
class Sales(Department):

    def __init__(self):

        self.strategy = "Growth via direct customer engagement"
        self.resources = ["Prospecting & qualification", "Negotiation"]
        self.structure = "Specialist"

    def execute(self, departments):

        # return "$$$"
        return "Distribution"
```

Org design: Business Development



Andrew Dumont
@AndrewDumont

If you use the terms Business Development and Sales interchangeably, you're doing it wrong.

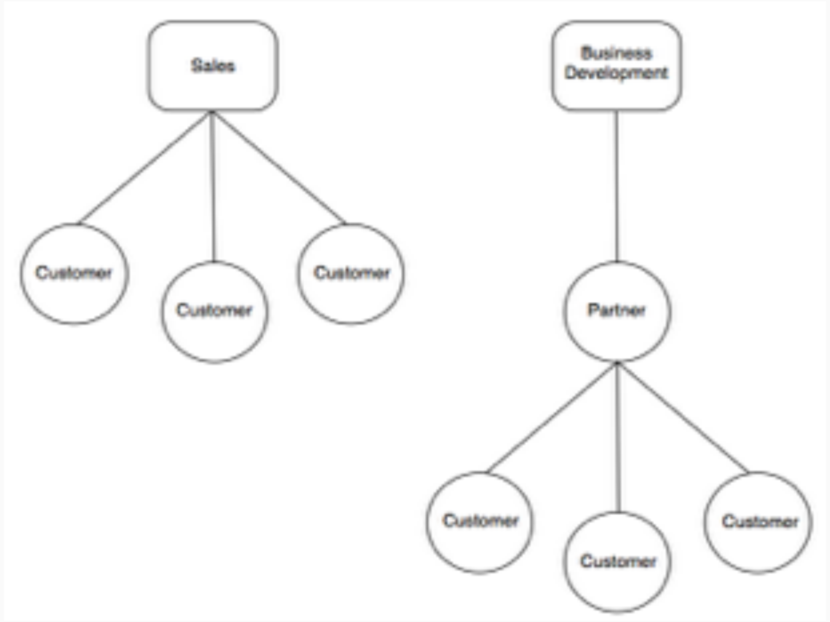
RETWEETS 57 LIKES 33

6:42 PM - 21 Mar 2013

Org design: Business Development

```
class BusinessDevelopment(Department):

    def __init__(self):

        self.strategy = "Leveraged growth via strategic business partnerships”
        self.resources = ["Partner research", “Relationships”, “Dealmaking”]
        self.structure = “Broker"
```

Org design: Business Development



The difference between Sales and Business Development (Dumont, 2014)

Org design: Business Development

```
class BusinessDevelopment(Department):

    def __init__(self):

        self.strategy = "Leveraged growth via strategic business partnerships”
        self.resources = ["Partner research", “Relationships”, “Dealmaking”]
        self.structure = “Broker"
```

Org design: Business Development

```
class BusinessDevelopment(Department):

    def __init__(self):

        self.strategy = "Leveraged growth via strategic business partnerships”
        self.resources = ["Partner research", “Relationships”, “Dealmaking”]
        self.structure = “Broker"

    def execute(self, departments):

        if Product in departments and Marketing in departments:
            return ["Lighthouse customer wins", "Leveraged platform integrations"]
```

Org design: Business Development

```
class BusinessDevelopment(Department):

    def __init__(self):

        self.strategy = "Leveraged growth via strategic business partnerships”
        self.resources = ["Partner research", “Relationships”, “Dealmaking”]
        self.structure = “Broker"

    def execute(self, departments):

        if Product in departments and Marketing in departments:
            return ["Lighthouse customer wins", "Leveraged platform integrations"]

        else:
            raise ValueError("\_(ツ)_/¯")
```

Org as an API for platform adoption

- Platform strategy
- Organizational structures and networks
- Organization design

#thankyou

@rchoi