Investigating a Drop in User Engagement:

1. SQL confirming nothing has really changed in the growth rate:

```
SELECT DATE_TRUNC('day',created_at) AS day,
       COUNT(*) AS all_users,
       COUNT(CASE WHEN activated_at IS NOT NULL THEN u.user_id ELSE NULL END)
AS activated_users
  FROM tutorial.yammer_users u
 WHERE created_at >= '2014-06-01'
   AND created_at < '2014-09-01'
 GROUP BY 1
 ORDER BY 1
```

2. SQL showing decrease in engagement for users who signed up more than 10 weeks prior:

```
SELECT DATE_TRUNC('week',z.occurred_at) AS "week",
       AVG(z.age_at_event) AS "Average age during week",
       COUNT(DISTINCT CASE WHEN z.user_age > 70 THEN z.user_id ELSE NULL END)
AS "10+ weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 70 AND z.user_age >= 63 THEN
z.user_id ELSE NULL END) AS "9 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 63 AND z.user_age >= 56 THEN
z.user_id ELSE NULL END) AS "8 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 56 AND z.user_age >= 49 THEN
z.user_id ELSE NULL END) AS "7 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 49 AND z.user_age >= 42 THEN
z.user_id ELSE NULL END) AS "6 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 42 AND z.user_age >= 35 THEN
z.user_id ELSE NULL END) AS "5 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 35 AND z.user_age >= 28 THEN
z.user_id ELSE NULL END) AS "4 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 28 AND z.user_age >= 21 THEN
z.user_id ELSE NULL END) AS "3 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 21 AND z.user_age >= 14 THEN
z.user_id ELSE NULL END) AS "2 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 14 AND z.user_age >= 7 THEN
z.user_id ELSE NULL END) AS "1 week",
       COUNT(DISTINCT CASE WHEN z.user_age < 7 THEN z.user_id ELSE NULL END)
AS "Less than a week"
  FROM (
        SELECT e.occurred_at,
               u.user_id,
               DATE_TRUNC('week',u.activated_at) AS activation_week,
               EXTRACT('day' FROM e.occurred_at - u.activated_at) AS
age_at_event,
               EXTRACT('day' FROM '2014-09-01'::TIMESTAMP - u.activated_at)
AS user_age
          FROM tutorial.yammer_users u
          JOIN tutorial.yammer_events e
            ON e.user_id = u.user_id
           AND e.event_type = 'engagement'
```

```
          AND e.event_name = 'login'
          AND e.occurred_at >= '2014-05-01'
          AND e.occurred_at < '2014-09-01'
        WHERE u.activated_at IS NOT NULL
      ) z

 GROUP BY 1
 ORDER BY 1
LIMIT 100
```

3. SQL showing drop in phone engagement rates:

```
SELECT DATE_TRUNC('week', occurred_at) AS week,
       COUNT(DISTINCT e.user_id) AS weekly_active_users,
       COUNT(DISTINCT CASE WHEN e.device IN ('macbook pro','lenovo
thinkpad','macbook air','dell inspiron notebook',
          'asus chromebook','dell inspiron desktop','acer aspire
notebook','hp pavilion desktop','acer aspire desktop','mac mini')
          THEN e.user_id ELSE NULL END) AS computer,
       COUNT(DISTINCT CASE WHEN e.device IN ('iphone 5','samsung galaxy
s4','nexus 5','iphone 5s','iphone 4s','nokia lumia 635',
       'htc one','samsung galaxy note','amazon fire phone') THEN e.user_id
ELSE NULL END) AS phone,
        COUNT(DISTINCT CASE WHEN e.device IN ('ipad air','nexus 7','ipad
mini','nexus 10','kindle fire','windows surface',
         'samsumg galaxy tablet') THEN e.user_id ELSE NULL END) AS tablet
  FROM tutorial.yammer_events e
 WHERE e.event_type = 'engagement'
   AND e.event_name = 'login'
 GROUP BY 1
 ORDER BY 1
LIMIT 100
```

4. SQL showing email clickthroughs are way down:

```
SELECT DATE_TRUNC('week', occurred_at) AS week,
       COUNT(CASE WHEN e.action = 'sent_weekly_digest' THEN e.user_id ELSE
NULL END) AS weekly_emails,
       COUNT(CASE WHEN e.action = 'sent_reengagement_email' THEN e.user_id
ELSE NULL END) AS reengagement_emails,
       COUNT(CASE WHEN e.action = 'email_open' THEN e.user_id ELSE NULL END)
AS email_opens,
       COUNT(CASE WHEN e.action = 'email_clickthrough' THEN e.user_id ELSE
NULL END) AS email_clickthroughs
  FROM tutorial.yammer_emails e
 GROUP BY 1
 ORDER BY 1
```

5. SQL showing that the problem has to do with digest emails in addition to mobile apps:

```
SELECT week,
       weekly_opens/CASE WHEN weekly_emails = 0 THEN 1 ELSE weekly_emails
END::FLOAT AS weekly_open_rate,
```

```
       weekly_ctr/CASE WHEN weekly_opens = 0 THEN 1 ELSE weekly_opens
END::FLOAT AS weekly_ctr,
       retain_opens/CASE WHEN retain_emails = 0 THEN 1 ELSE retain_emails
END::FLOAT AS retain_open_rate,
       retain_ctr/CASE WHEN retain_opens = 0 THEN 1 ELSE retain_opens
END::FLOAT AS retain_ctr
  FROM (
SELECT DATE_TRUNC('week',e1.occurred_at) AS week,
       COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e1.user_id ELSE
NULL END) AS weekly_emails,
       COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e2.user_id ELSE
NULL END) AS weekly_opens,
       COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e3.user_id ELSE
NULL END) AS weekly_ctr,
       COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e1.user_id
ELSE NULL END) AS retain_emails,
       COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e2.user_id
ELSE NULL END) AS retain_opens,
       COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e3.user_id
ELSE NULL END) AS retain_ctr
  FROM tutorial.yammer_emails e1
  LEFT JOIN tutorial.yammer_emails e2
    ON e2.occurred_at >= e1.occurred_at
   AND e2.occurred_at < e1.occurred_at + INTERVAL '5 MINUTE'
   AND e2.user_id = e1.user_id
   AND e2.action = 'email_open'
  LEFT JOIN tutorial.yammer_emails e3
    ON e3.occurred_at >= e2.occurred_at
   AND e3.occurred_at < e2.occurred_at + INTERVAL '5 MINUTE'
   AND e3.user_id = e2.user_id
   AND e3.action = 'email_clickthrough'
 WHERE e1.occurred_at >= '2014-06-01'
   AND e1.occurred_at < '2014-09-01'
   AND e1.action IN ('sent_weekly_digest','sent_reengagement_email')
 GROUP BY 1
       ) a
 ORDER BY 1
```

Follow through:

After investigation, it appears that the problem has to do with mobile use and digest emails. The intended action here should be clear: notify the head of product (who approached you in the first place) that the problem is localized in these areas and that it’s worth checking to make sure something isn’t broken or poorly implemented.